

**O`ZBEKISTON RESPUBLIKASI
OLIIY VA O`RTA MAXSUS TA'LIM VAZIRLIGI
QARSHI DAVLAT UNIVERSITETI**

E.A.Eshboyev, F.Yu.Shodiyev, F.G.Qilicheva

C++ TILIDA DASTURLASH

(Mustaqil o`rganuvchilar uchun amaliy qo`llanma)

I-QISM. STRUKTURALI DASTURLASH

MUNDARIJA

Soʻz boshi	3
I Bob. Oʻzgaruvchilar, ifodalar va chiziqli dasturlar.....	5
1.1. Oʻzgaruvchilarning turlari va ifodalarning yozilishi	5
1.2. Chiziqli algoritmlar va ular asosida dasturlar qurish	11
1.3. Mantiqiy ifodalar	23
1.4. Bobga doir qiyinroq masalalar	28
II Bob. Tarmoqlanuvchi algoritmlar va ularni dasturlash	34
2.1. Tarmoqlanish va shartsiz oʻtish jarayonlari	34
2.2. Tanlash operatori	40
2.3. Bobga doir qiyinroq masalalar	44
III Bob. Takrorlanuvchi algoritmlar va ularni dasturlash.....	48
3.1. for operatoridan foydalanib takrorlash jarayonlarini tashkil etish	48
3.2. while va do-while operatorlaridan foydalanish.....	55
3.3. Bobga doir qiyinroq masalalar	62
IV Bob. Massivlar.....	74
4.1. Dasturlashda bir oʻlchovli massivlarni qoʻllash.....	74
4.2. Koʻp oʻlchovli massivlar	90
4.3. Bobga doir qiyinroq masalalar	100
V Bob. Belgilar va satrlar	110
5.1. Belgilar	110
5.2. Satrlar	112
5.3. Bobga doir qiyinroq masalalar	123
VI Bob. Funksiyalar.....	126
6.1. Sonli parametrli funksiyalar	128
6.2. Funksiyalarda koʻrsatkichlar va murojaatlar.....	138
6.3. Rekursiya.....	142
6.4. Bobga doir qiyinroq masalalar	145
VII Bob. Fayllar bilan ishlash.....	150
7.1. Sonli va satrli fayllar	151
7.2. Matnli fayllar	162
7.3. Bobga doir qiyinroq masalalar	168
VIII Bob. Funksiyalarda maʼlumotlarning murakkab turlari	176
8.1. Bir va ikki oʻlchovli massivlar bilan ishlashda funksiyalardan foydalanish	176
8.2. Satrlar va fayllarga doir masalalarni yechishda funksiyalardan foydalanish	182
8.3. Strukturalar	188
8.4. Bobga doir qiyinroq masalalar	195
ILOVA: <i>math.h</i> kutubxonasining funksiyalari	202
Foydalanilgan adabiyotlar va internet resurslari:	204

Soʻz boshi

Dastur mashina kodlarining ketma-ketligi boʻlib, u aniq bir hisoblash jarayonini amalga oshiradi. Shuning uchun ham dasturiy vositalarni ishlab chiqishni soddalashtirish maqsadida koʻplab dasturlash tillari yaratilgan. Mavjud dasturlash tillari shartli ravishda ikki toifaga ajratiladi:

- quyi bosqichli dasturlash tillari;
- yuqori bosqichli dasturlash tillari.

Quyi bosqichdagi dasturlash tillariga *Assembler* turidagi tillar kiradi. Bu tillar nisbatan qisqa va tezkor bajariluvchi kodlarni yaratish imkonini beradi[1]. Ammo, *Assembler* tilida dastur tuzish qiyin (zahmatli), ya'ni uzoq davom etadigan jarayondir.

Dasturlar tuzishni yengillashtirish maqsadida yuqori bosqichli dasturlash tillari yaratilgan boʻlib, ularda tabiiy tillarda mavjud imkoniyatlarning chekli qismidan foydalanib dasturlar tuziladi. Yuqori bosqichli tillardagi operatorlar, berilganlarning turlari, oʻzgaruvchilar, hamda dastur yozishning turli usullari bu bosqichdagi tillarning ifodalash imkoniyatini oshiradi, ya'ni yaratilgan dasturlarning «oʻqimishli» boʻlishini taʼminlaydi. Yuqori bosqich tillariga *Basic*, *Pascal*, *C* kabilarni misol qilib keltirish mumkin. Kompyuterlar (hisoblash mashinalari)ning takomillashuvi, hamda tarmoq texnologiyalarining rivojlanishi yuqori bosqich tillarining yangi turlari yuzaga kelishiga va ayrimlarining «*iste'moldan chiqib*» qolishiga sabab boʻlmoqda. Hozirda kunda keng tarqalgan tillar *Object Pascal*, *C++*, *C#*, *Php*, *Java*, *Asp* kabilar hisoblanadi. Jumladan, *C* tilining avlodi sifatida *Byarn Straustrop* tomonidan 1980 yilda yaratilgan *C++* tili strukturali va obʼyektga yoʻnaltirilgan dasturlash texnologiyasiga tayangan holda dastur yaratish imkoniyatini berdi.

Ushbu qoʻllanma *C++* tilida strukturali dasturlashni oʻrganishga moʻljallangan boʻlib, mualliflarning bir necha yillik izlanishlari natijasi sifatida yozilgan. Qoʻllanmada har bir boʻlim uchun qisqacha nazariy maʼlumotlar, dastlabki 4 ta bobda esa qoʻyilgan namunaviy masalalar uchun blok-sxema koʻrinishidagi algoritm va izohlar bilan berilgan dastur matni keltirilgan. Qolgan boblarda qoʻyilgan masalalar uchun tuzilgan dasturlar izohlar berish yoʻli bilan tushuntirilgan. Shuni taʼkidlash lozimki, qoʻllanmada mavjud boblarning har bir boʻlimida qiyinlik darajasiga qarab toifalarga ajratilgan masalalar keltirilgan. Berilganlarga kiritiladigan qiymatlarning kiritilishini toʻgʻri amalga oshirish uchun

masalalarning matnida berilganlar chegarasi qavs ichida keltirilgan. Bu kabi qavs ichida yozilgan shartlarga dasturga qiymatlar kiritishda e'tibor qaratish kerak. Bundan tashqari, masalalar matnida ilk bor uchraydigan tushunchalarga qavs ichida izohlar keltirilgan.

Har bir bobning oxiridan bobga doir qiyinroq toifadagi masalalar o'rin olgan bo'lib, ular o'rganuvchilarning bilimlarini yanada mustahkamlashga xizmat qiladi. Mazkur kitob mustaqil o'rganuvchilar, va talabalarning C++ tilida dasturlashni qisqa muddat ichida o'rganishlari uchun amaliy qo'llanma bo'lib xizmat qiladi.

I Bob. O'zgaruvchilar, ifodalar va chiziqli dasturlar

1.1.O'zgaruvchilarning turlari va ifodalarning yozilishi

Dastur bajarilishi mobaynida berilganlarni saqlab turish uchun o'zgaruvchilar va o'zgarmlardan foydalaniladi.

O'zgaruvchi–dastur obyekti bo'lib, xotiradagi bir nechta yacheykalarni egallaydi va berilganlarni saqlash uchun xizmat qiladi. O'zgaruvchi nomga, o'lchamga va boshqa atributlarga – ko'rinish sohasi, amal qilish vaqti va boshqa xususiyatlarga ega bo'ladi. O'zgaruvchilarni ishlatish uchun ular albatta e'lon qilinishi kerak. E'lon natijasida o'zgaruvchi uchun xotiradan ma'lum adresga ega soha zahiralanadi, soha o'lchami esa o'zgaruvchining turiga bog'liq bo'ladi. Shuni qayd etish zarurki, bitta turga tegishli o'zgaruvchiga turli apparat platformalarda xotiradan turlicha joy ajratilishi mumkin.

O'zgaruvchi e'loni uning turini aniqlovchi kalit so'z bilan boshlanadi va '=' belgisi orqali boshlang'ich qiymat beriladi (o'zgaruvchiga boshlang'ich qiymat berilmasa ham bo'ladi). Bitta kalit so'z bilan bir nechta o'zgaruvchilarni e'lon qilish ham mumkin. Buning uchun o'zgaruvchilar bir-biridan ',' belgisi bilan ajratiladi. O'zgaruvchilar e'loni ';' belgisi bilan tugallanadi. O'zgaruvchi nomi 255 ta belgidan oshmasligi kerak.

Quyida C++ tilining tayanch turlari, ularning baytlardagi o'lchamlari va qiymatlarining chegaralarini keltiramiz:

<i>Tur nomi</i>	<i>Baytlardagi o'lchami</i>	<i>Qiymat chegarasi</i>
Unsigned short int	2	0..65535
Short int	2	-32768..32767
Unsigned long int	4	0..42949667295
Long int	4	-2147483648..2147483647
int(16 razryadli)	2	-32768.. 32767
Int (32 razryadli)	4	-2147483648..2147483647
Unsigned int (16 razryadli)	2	0..65535
Unsigned int (32 razryadli)	4	0..42949667295
Unsigned char	1	0..255
Char	1	-128.. 127
Float	4	1.2E-38..3.4E+38
Double	8	2.2E-308..1.8E+308

Long double (32 razryadli)	10	3.4E-4932..-3.4E+4932
Bool	1	true yoki false
Void	2 ёки 4	-

1.1-jadval. Tayanch turlar.

Endi 1.1-jadvalda keltirilgan turlar tavsifini keltiramiz:

Butun son turlari. Butun son qiymatlarni qabul qiladigan o'zgaruvchilar *int* (butun), *short* (qisqa), va *long* (uzun) kalit so'zlari bilan e'lon qilinadi. O'zgaruvchi qiymatlari ishorali bo'lishi yoki *unsigned* kalit so'zi bilan ishorasiz son sifatida qaralishi mumkin.

Belgi turi. Belgi turidagi o'zgaruvchilar *char* kalit so'zi bilan e'lon qilinadi va ular o'zida belgining *ASCII* jadvalidagi kodini saqlaydi. Belgi turidagi qiymatlar nisbatan murakkab bo'lgan tuzilmalar – satrlar, belgilar massivi va hokazolarni hosil qilishda ishlatiladi.

Haqiqiy son turi. Haqiqiy sonlar *float* va *double* kalit so'zlari bilan e'lon qilinadi. Bu turdagi o'zgaruvchilar uchun mos ravishda xotiradan 4 va 8 bayt joy ajratiladi, hamda sonni saqlash *<ishora><tartib><mantissa>* qolipida bo'ladi. Agar kasrli son juda katta (kichik) qiymatlarni qabul qiladigan bo'lsa, u xotiradan 10 bayt joy egallaydi va ikkilangan anqlik ko'rinishida saqlanadi. U *long double* kalit so'zi bilan e'lon qilinadi.

Mantiqiy tur. Bu turdagi o'zgaruvchi *bool* kalit so'zi bilan e'lon qilinadi va u xotiradan bir bayt joy egallaydi, hamda 0(*yolg'on*) yoki 1(*rost*) qiymatlaridan birini qabul qilishi mumkin.

Void turi. *Void* turidagi dastur obyekti hech qanday qiymatga ega bo'lmaydi va bu turdan qurilmaning til sintaksisiga mos kelishini ta'minlash uchun ishlatiladi. Masalan, C++ tili sintaksisi funksiya qiymat qaytarishini talab qiladi. Agar funksiyaning qiymat qaytarishi talab etilmasa, u void kalit so'zi bilan e'lon qilinadi.

O'zgaruvchilarning e'lon qilinishiga doir misollar:

Int *a*=-5, *A*=15;

float *abc*=36.7, *d*;

double *kasr_son*;

bool *b*,*t*;

char *BELGI*='@';

Void *mening_funksiyam()*; /*funksiya qaytaradigan qiymat inobatga olinmaydi */

Mavzuga doir masalalar

A toifa

1.001 Quyidagi sonlar dasturlash tilida yozilsin (kasr qismida to'rtta raqam ko'rsatilsin).

a) $4!$; b) $CLXV$; v) $-0,8(3)$; g) $\frac{7}{11}$; d) $-\frac{102}{155}$;

1.002 Quyidagi sonlar 10 darajasidan foydalanilmagan ko'rinishda yozilsin:

a) $-0.00123E+4$; b) $456E-3$; v) $21E+2$; g) $-321E-3$;

1.003 C++ dasturlash tili nuqtai nazaridan 55 va 55.0, hamda 60 va 6E1 sonlarining bir-biridan farqi bormi? Butun sonlar haqiqiy sonlardan qanday belgisiga qarab farqlanadi?

Quyidagi sonlar dasturlash tilida yozilsin(1.004-1.015)

1.004 a) $(1+x)^2$; b) $\sqrt{1+x}$; v) $|ax+b|$; g) $\sin 5$; d) $\cos^3 x^5$;

1.005 ($x>0$):

a) x^{-1} ; b) x^4 ; v) x^{-2} ; g) x^5 ;

1.006 a) $a+bx+cyz$; b) $[(ax-b)x+c]x-d$; v) $\frac{ab}{c} + \frac{c}{ab}$;

1.007 a) $y=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}$; b) $f=6,673 \cdot 10^{-8} \frac{m_1 \cdot m_2}{r^2}$;

v) $b=e^{|x-y|} + \ln(1+e) \log_2 \lg 5$.

1.008 $7,02 \cdot 10^{-5} \cdot 2^{3/7} + 0,3\sqrt{2,5} + e^{1,5}$

1.009 $\sqrt[4]{\frac{2}{3} + \sqrt{\frac{3}{2}}} + \left| 2^3 - \sin 15^\circ \right|$

1.010 $\frac{1}{1-0,3} \cdot \ln \left| \frac{1+2^{0,3}}{1-3^{0,2}} \right| + \frac{\sqrt[7]{12}}{0,8}$

1.011 $\sqrt[3]{0,7+(0,3)^{0,4}} - \sqrt[3]{8^{0,2}} - e^{-1/3}$

1.012 $\frac{2}{\pi} \arctg \frac{1}{\pi} + \sqrt[3]{1-0,2^3}$

1.013 $\left(1 + \frac{1}{0,3^{12}} \right)^3 + e^{\frac{1}{3-\lg^2 25}}$

1.014 $\frac{1}{2\sqrt[4]{0,3}} \cdot \ln \frac{\sqrt[5]{3} - \sqrt[3]{5}}{2} - e^{0,7}$

- 1.015** $\sin\left(12 + e^{0,7 + \cos\frac{\pi}{8}} + \frac{1}{\sqrt{0,3^3}}\right)$
- 1.016** Dasturlash tillarida $(1/6)*6-1=$ ifodaning qiymati nolmi?
- 1.017** Nima uchun dasturlash tillarida formulalar bir chiziqqa keltirib yoziladi? Ko'paytirish belgisini har doim yozish shartmi (*masalan, $b*c$ yoki bc*)?
- 1.018** Dasturlash tilida x gradusning sinusi qanday yozilishi ko'rsatilsin?
- 1.019** Hisoblansin: $24/(3*4) - 24/3/4 + 24/3*4$
- 1.020** Quyidagi ifodani hisoblashda nechta amal bajariladi? Amallar sonini qanday kamaytirish mumkin?
 $(x+1/2)*(y+7/10)-3/4;$
- 1.021** d -o'zgaruvchiga quyidagi qiymatlarni o'zlashtiruvchi buyruq dasturlash tilida yozilsin.
a) berilgan x, y, z sonlarining o'rta arifmetigini;
b) (x_1, y_1) va (x_2, y_2) nuqtalar orasidagi masofani.
- 1.022** Quyidagi operatorlar bajarilgandan keyin x o'zgaruvchi qanday qiymat qabul qiladi? $x=123;$ $x=x+9.$
- 1.023** Dasturlash tilida t o'zgaruvchi qiymatining ishorasini o'zgartiruvchi qiymat berish operatori yozilsin.
- 1.024** Quyidagi operatorlar bajarilganidan keyin x va y o'zgaruvchilarning qiymati qanday bo'ladi:
 $x=5;$ $y=9;$ $x=y;$ $y=x.$
- 1.025** Ikkita x va y o'zgaruvchilar qiymatlarini o'zaro almashtiruvchi operatorlar ketma-ketligi yozilsin:
- 1.026** Ifodalarning turlari (butun yoki haqiqiy) aniqlansin:
a) $3+0.0;$ b) $28/4;$ v) $\sin(0);$
- 1.027** Quyidagi o'zlashtirish operatoridan qaysi biri to'g'ri (yoki noto'g'ri) va nima uchun? Bu yerda y -haqiqiy, n -esa butun o'zgaruvchi.
a) $y=n+1;$ b) $n=y-1;$ v) $n=4.0;$ g) $n:=n/2;$
- 1.028** Nima uchun dasturlash tillarida funksiya argumenti qavs ichida yoziladi (*masalan, $\cos(5)$, lekin $\cos 5$ emas*).
- 1.029** Dasturlash tillarida $(1/6)*6-1=$ ifodaning qiymati nolmi?
- 1.030** Nima uchun dasturlash tillarida formulalar bir chiziqqa keltirib yoziladi? Ko'paytirish belgisini har doim yozish shartmi ($b*c$ yoki bc)?

B toifa

Quyidagi ifodalar dasturlash tilida yozilsin (1.031-1.060)

1.031 $(x>0)da, a) x^{100}; b) 2^{1+x}; c) x^{\sqrt{2}}; d) \sqrt[3]{1+x};$

1.032 $a) \frac{x+y}{a_1} \cdot \frac{a_2}{x-y}; b) 10^4 \alpha - 5 \frac{2}{7} \beta; c) \left(1 + \frac{x}{2!} + \frac{y}{3!}\right) / \left(1 + \frac{2}{3+xy}\right);$

1.033 $a) \sin 8; b) \cos^2 x^8; c) \operatorname{tg} x; d) \log_2 \frac{x}{5};$

1.034 $s = a^2 \operatorname{tg} \frac{b}{2} \operatorname{tg} \frac{a}{2} - \sqrt{\frac{1+\cos a}{b}};$

1.035 $\left(\frac{3}{4}\right)^5 \cdot \left(\frac{4}{0,3}\right)^{0,2} + \ln^2(12 - 0,3^3);$

1.036 $\frac{(0,2)^{0,5} - 1}{12 + \lg 3} - 2 \cdot 10^{3,721} + \operatorname{arctg} \sqrt{68};$

1.037 $\ln(2e^{0,4} + 0,3)^{7/5} + \frac{1}{\sqrt[8]{3}} - \sin 322^0;$

1.038 $\frac{4,5^{5,321} - \cos \ln^2(13 - 0,3^4)}{\sqrt{|12 - 13^2|}^6};$

1.039 $\sqrt[6]{3 + \log_2(4 + 3,3 \cdot 10^{-12})};$

1.040 $\sqrt{3 + \sqrt[3]{4 + \sqrt[4]{5}}} + \sin \frac{1}{\sqrt{\cos \frac{\pi}{8}}};$

1.041 $\frac{0,15 \cdot \sqrt[3]{12} e^{\sin \frac{\pi}{7}} + \cos^3 \frac{\pi}{12}}{\ln \lg^{12} 64,54};$

1.042 $\frac{12 + \left[3,3 + 2[4 + 0,7 \cdot 0,8]^3\right]}{0,7^{0,6}};$

1.043 x, y, z o'zgaruvchilar qiymatlari shunday almashtirilsinki, natijada x o'zgaruvchi y ning qiymatini, y va z ning qiymatini, z o'zgaruvchi esa x o'zgaruvchining avvalgi qiymatini qabul qilsin.

1.044 Musbat x sonining kasr qismini d o'zgaruvchiga beradigan o'zlashtirish operatori yozilsin.

1.045 Ifodalar qiymati hisoblansin:

- a) 20/6; b) 20%6; v) 20/4; g) 20%4;
- 1.046** Boshqa amallar va standart funksiyalar orqali butunga bo'lish amali ifodalansin:
a) 2/5; b) 2%5; vj) 123/0; g) 3.0%3
- 1.047** Quyidagi ifoda uchun amallarni bajarish tartibi ko'rsatilsin:
-a%b +d/b*c.
- 1.048** Ifodalarning qiymatlari hisoblansin:
a) 3*7/2%7/3-(int) sin(1); b) sqrt(int (4,987)*pow(2,-3));
- 1.049** Ifodalarning turlari (butun yoki haqiqiy) aniqlansin.
a) 1+0.0; b) 20/4; v) (float) 4;
g) pow (5,0); d) sqrt (16); ye) sin (0);
j) (int) -2; z) pow(7,-3);
- 1.050** C++ tilida haqiqiy turdagi o'zgaruvchi (aytaylik x) qiymati faqat haqiqiy son bo'lishi kerak va shu bilan birgalikda bu o'zgaruvchiga butun qiymat o'zlashtirilishi ham mumkin (masalan, x:=7). Bu zidlikni qanday bartaraf qilish mumkin?
- 1.051** Quyidagi o'zlashtirish operatoridan qaysi biri to'g'ri yoki noto'g'ri? Bu yerda y-haqiqiy, n-esa butun o'zgaruvchi.
a) y=n+1; b) n=y-1; v) n=4.0;
g) y= (int) y; d) n=(float) n/2; e) y=y/2;
f) n=n/2; j) n=sqrt(n);
- 1.052** Quyidagi o'zlashtirish operatori to'g'rimi? Javob asoslansin.
a) k=k%3+k*cos(0); b) x=x*2/6+x/4;
- 1.053** Butun turdagi h o'zgaruvchiga, berilgan k musbat sonning oxiridan uchinchi raqami o'zlashtirilsin (masalan, agar k=130985 bo'lsa, h=9).
- 1.054** Butun turdagi d o'zgaruvchiga haqiqiy musbat x sonining kasr qismining birinchi raqami o'zlashtirilsin (masalan, agar x=32.597 bo'lsa, d=5).
- 1.055** Uch xonali butun k son raqamlarini yig'indisini s butun o'zgaruvchiga o'zlashtirilsin.
- 1.056** Agar hozir sutkaning k-sekundi bo'lsa, shu paytgacha necha soat (h) va minut (m) o'tganligi hisoblansin (masalan, agar k=13257=3*3600+40*60+57 bo'lsa, h=3 va m=40).
- 1.057** Soat milining sutka boshidagi holati bilan h soat, m minut va s sekunddagi holatlari orasidagi f burchak (graduslarda) aniqlansin. ($0 \leq h \leq 11, m \geq 0, s \leq 59$).
- 1.058** Soat milining sutka boshiga nisbatan f gradus burilishiga mos

keluvchi h soat, m minut va s sekund topilsin ($0 \leq f \leq 360^\circ$, f – haqiqiy son).

1.059 1-yanvar kuni dushanba bo'lsa, kabisa bo'lmagan biror yilning k -kuni ($0 \leq k \leq 365$) haftaning qaysi kuniga to'g'ri kelishi aniqlansin va bu qiymatni n butun o'zgaruvchiga o'zlashtirilsin ($0 \leq n \leq 7$).

1.060 1-yanvar kuni juma bo'lsa, kabisa bo'lmagan biror yilning k -kuni ($0 \leq k \leq 365$) haftaning qaysi kuniga to'g'ri kelishi aniqlansin va bu qiymatni n butun o'zgaruvchiga o'zlashtirilsin ($0 \leq n \leq 7$).

1.2. Chiziqli algoritmlar va ular asosida dasturlar qurish

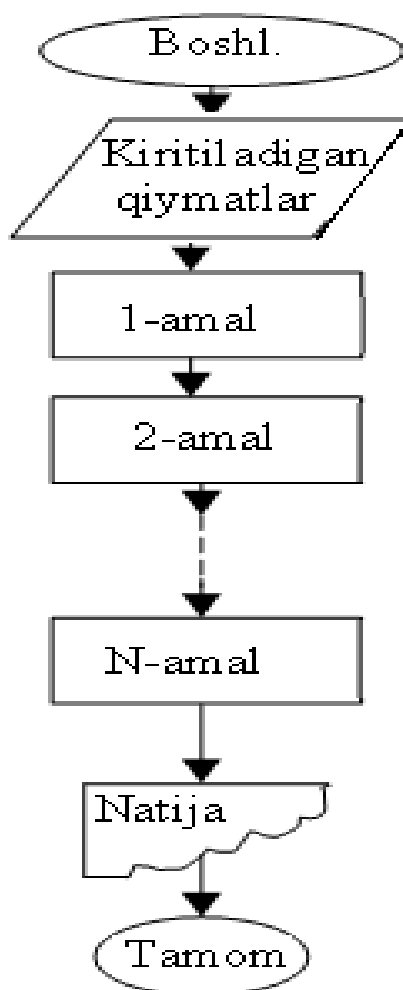
Kompyuter uchun tuzilgan algoritm ijrochisi-bu kompyuterdir. Biror dasturlash tilida yozilgan algoritm kodlashtirilgan oddiy ko'rsatmalar ketma-ketligiga o'tadi va mashina tomonidan avtomatik ravishda bajariladi. Metodik nuqtai-nazardan qaraganda algoritmning birinchi ijrochisi sifatida o'quvchi (talaba) o'zini tanlashi mumkin. O'quvchi tomonidan biror masalani yechish algoritmi tuzilganda bu algoritmning to'g'ri natija berishini tekshirish juda muhimdir. Buning yagona usuli o'quvchi tomonidan algoritmni turli boshlang'ich berilganlarda qadamma - qadam bajartirib (ijro etib) ko'rishidir. Algoritmni bajarish natijasida xatolar aniqlanadi va bartaraf etiladi. Ikkinchi tomonidan, masalani yechishga qiynalayotgan o'quvchi uchun tayyor algoritmni bajarish – masalani yechish yo'llarini tushunishga xizmat qiladi.

Algoritmni shartli ravishda quyidagi turlarga ajratish mumkin:

- *chiziqli algoritmlar*;
- *tarmoqlanuvchi algoritmlar*;
- *takrorlanuvchi yoki siklik algoritmlar*;
- *ichma-ich joylashgan siklik algoritmlar*;
- *rekurrent (rekursiv) algoritmlar*;
- *takrorlanishlar soni oldindan noma'lum algoritmlar*;
- *ketma-ket yaqinlashuvchi algoritmlar*.

Algoritmni bajarilish tartibi navbat bilan, ketma-ket tarzda amalga oshsa – bunday algoritmga *chiziqli algoritm* deyiladi.

Chiziqli algoritmning blok-sxemasini tasvirlashning umumiy ko'rinishi quyidagicha bo'ladi:



C++ tilida o'zgaruvchilarga qiymat berish yoki ifodalarni qiymatlash uchun “=” amalidan foydalaniladi. Uning sintaksisi quyidagicha:

`<o'zgaruvchi>=<qiymat yoki ifoda>;`

O'zgaruvchilarga klaviaturadan qiymat kiritish va natijani ekranga chiqarish uchun mos ravishda “*iostream.h*” kutubxonasida joylashgan “*cin*” va “*cout*” oqimli o'qish yozish funksiyalaridan foydalaniladi. Ularning sintaksislari quyidagicha:

`cin>>o'zgaruvchi;`

`cout<<o'zgaruvchi yoki ifoda;`

O'zgaruvchilar soni nechta bo'lsa, “>>” (“*get*”), “<<” (“*put*”) amallari ham shuncha marta qo'llaniladi.

Bulardan tashqari C++ tilida oqimdan formatli o'qish va yozishni amalga oshirishga mo'ljallangan funksiyalar ham mavjud. Quyida ular bilan tanishamiz.

Bu funksiyalar – *scanf()* va *printf()* lar bo'lib, C tilidan vorislik bilan olingan. Ulardan foydalanish uchun «*stdio.h*» sarlavha faylini dasturga qo'shish kerak bo'ladi.

Formatli o'qish funksiyasi *scanf()* quyidagi,

*int scanf(const char * <format> [<adres>, ...]);*

prototipga ega. Formatlash satri – <format> belgilar satri bo'lib, u uchta toifaga bo'linadi:

1. To'ldiruvchi belgilar — bu 'bo'sh joy', '\t', '\n' belgilari. Bu belgilar formatlash satridan o'qiladi, lekin o'zgaruvchilarda saqlanmaydi.

2. To'ldiruvchi belgilardan farqli belgilar – bu '%' belgisidan tashqari, qolgan barcha *ASCII* belgilari hisoblanadi. Bu belgilar ham formatlash satridan o'qiladi, lekin o'zgaruvchilarda saqlanmaydi.

3. Format aniqlashtiruvchilar – oqim maydonidagi belgilarni ko'rib chiqish, o'qish va adresi bilan berilgan o'zgaruvchilar turiga mos ravishda almashtirish jarayonini boshqaradi. Har bir format aniqlashtiruvchisiga bitta o'zgaruvchi adresi mos kelishi kerak. Agar format aniqlashtiruvchilari soni o'zgaruvchilardan ko'p bo'lsa, natija qanday bo'lishini oldindan aytib bo'lmaydi. Aks holda, ya'ni o'zgaruvchilar soni ko'p bo'lsa, ortiqcha o'zgaruvchilar inobatga olinmaydi.

Format aniqlashtiruvchisi quyidagi ko'rinishga ega:

%[][<kenglik>][F|N][h|1|L]<tur belgisi>*

Format aniqlashtiruvchisi '%' belgisidan boshlanadi va undan keyin 1.2-jadvalda keltirilgan shart yoki shart bo'lmagan komponentalar keladi:

Komponenta	Bo'lishi shart yoki yo'q	Vazifasi
[*]	yo'q	Navbatdagi ko'rib chiqilayotgan maydon qiymatini o'zgaruvchiga o'zlashtirmaslik belgisi. Kirish oqimidagi maydon ko'rib chiqiladi, lekin o'zgaruvchida saqlanmaydi.
[<kenglik>]	yo'q	Maydon kengligining aniqlashtiruvchisi. O'qiladigan belgilarning maksimal sonini aniqlaydi. Agar oqimda to'ldiruvchi belgi yoki almashtirilmaydigan belgi uchrasa funksiya nisbatan kam sondagi belgilarni o'qishi mumkin.
[F N]	yo'q	O'zgaruvchi ko'rsatkichining (adresining) modifikatori: <i>F</i> - far pointer; <i>N</i> - near pointer.
[h 1 L]	yo'q	Argument turining modifikatori. <tur

		<i>belgisi</i> > bilan aniqlangan o'zgaruvchi-ning qisqa (<i>short - h</i>) yoki uzun (<i>long - 1,L</i>) ko'rinishini aniqlaydi.
<tur belgisi>	ha	Oqimdagi belgilarni almashtiriladigan tur belgisi.

1.2-jadval. Format aniqlashtiruvchilari va ularning vazifasi.

Oqimdagi belgilarni almashtiriladigan tur belgisining qabul qilishi mumkin bo'lgan belgilari quyidagi jadvalda keltirilgan:

Berilganlar turi	Kutilayotgan qiymat	Argument turi
Son turidagi argument		
d,D	O'nlik butun	Butunga ko'rsatkich (<i>int * arg</i> yoki <i>long * arg</i>)
E,e	Suzuvchi nuqtali son	<i>float</i> turiga ko'rsatkich (<i>float * arg</i>)
f	Suzuvchi nuqtali son	<i>float</i> turiga ko'rsatkich (<i>float * arg</i>)
G,g	Suzuvchi nuqtali son	<i>float</i> turiga ko'rsatkich (<i>float * arg</i>)
O	Sakkizlik son	Butunga ko'rsatkich (<i>int * arg</i>)
O	Sakkizlik son	Butunga ko'rsatkich (<i>long * arg</i>)
I	O'nlik, sakkizlik va o'n oltilik butun son	Butunga ko'rsatkich (<i>int * arg</i>)
I	O'nlik, sakkizlik va o'n oltilik butun son	Butunga ko'rsatkich (<i>long * arg</i>)
U	Ishorasiz o'nlik son	<i>unsigned int</i> turiga ko'rsatkich (<i>unsigned int * arg</i>)
U	Ishorasiz o'nlik son	<i>unsigned long</i> turiga ko'rsatkich (<i>unsigned long * arg</i>)
X	O'n oltilik son	Butunga ko'rsatkich (<i>int * arg</i>)
X	O'n oltilik son	Butunga ko'rsatkich (<i>int * arg</i>)
Belgilar		
s	Satr	Belgilar massivga ko'rsatkich (<i>char * arg</i>)
c	Belgi	Belgiga ko'rsatkich (<i>char * arg</i>).

		Belgi uchun maydon kengligi berilishi mumkin (masalan, %4). <i>N</i> belgidan tashkil topgan belgilar massiviga ko'rsatkich (<i>char arg[N]</i>).
%	'%' belgisi	Hech qanday almashtirishlar bajarilmaydi, '%' belgisi saqlanadi.
Ko'rsatkichlar		
n	int ko'rsatkich (int * arg)	% <i>n</i> argumentigacha muvaffaqiyatli o'qilgan belgilar soni aynan shu int ko'rsatkichi bo'yicha adresda saqlanadi.
p	YYYY:ZZZZ yoki ZZZZ ko'rinishidagi o'n oltilik	Obyektga ko'rsatkich (<i>far*yoki near*</i>)

1.3-jadval. Almashtiriladigan tur belgilari.

Formatli yozish funksiyasi *printf()* esa quyidagi,

*int printf(const char * <format>[<argument>,...]);*

prototipga ega bo'lib, bu funksiya standart oqimga formatlashgan chiqarishni amalga oshiradi. Funksiya argumentlar ketma-ketligidagi har bir argument qiymat qabul qiladi va unga <format> satridagi mos format aniqlashtiruvchisini qo'llaydi va oqimga chiqaradi. Har bir format aniqlashtiruvchisiga bitta o'zgaruvchi adresi mos kelishi kerak. Agar format aniqlashtiruvchilari soni o'zgaruvchilardan ko'p bo'lsa, natija nima bo'lishini oldindan aytib bo'lmaydi. Aks holda, ya'ni o'zgaruvchilar soni ko'p bo'lsa, ortiqcha o'zgaruvchilar inobatga olinmaydi. Agar oqimga chiqarish muvaffaqiyatli bo'lsa, funksiya chiqarilgan baytlar sonini, aks holda *EOF* qiymatini qaytaradi.

printf() funksiyasining <format> satri argumentlarni almashtirish, formatlash va chiqarish jarayonini boshqaradi va u ikki turdagi obyektlardan tashkil topgan.

- oqimga o'zgarishsiz chiqariladigan oddiy belgilar;
- argumentlar ro'yxatidan tanlanadigan argumentga qo'llaniladigan format aniqlashtiruvchilari.

U quyidagi ko'rinishga ega:

% [<bayroq>][<kenglik>] [.<xona>][F|N|h|l|L] <tur belgisi>;

Format aniqlashtiruvchisi ‘%’ belgisidan boshlanadi va undan keyin 1.4-jadvalda keltirilgan shart yoki shart bo’lmagan komponentalar keladi:

Komponenta	Bo’lishi shart yoki yo’q	Vazifasi
[bayroq]	yo’q	Bayroq belgilari. Chiqarilayotgan qiymatni chapga yoki o’nga tekislashni, sonning ishorasini, o’nlik kasr nuqtasini, oxirdagi nollarni, sakkizlik va o’n oltilik sonlarning alomatlarini chop etishni boshqaradi. Masalan, ‘-‘ bayrog’i qiymatni ajratilgan o’ringa nisbatan chapdan boshlab chiqarishni va kerak bo’lsa o’ngdan bo’sh joy bilan to’ldirishni bildiradi, aks holda chap tomondan bo’sh joylar bilan to’ldiradi va davomiga qiymat chiqariladi.
[<kenglik>]	yo’q	Maydon kengligining aniqlashtiruvchisi. Chiqariladigan belgilarning minimal sonini aniqlaydi. Zarur bo’lsa qiymat yozilishidan ortgan joylar bo’sh joy bilan to’ldiriladi.
[.<xona>]	yo’q	Aniqlik. Chiqariladigan belgilarning maksimal sonini ko’rsatadi. Sonlar uchun—chiqariladigan raqamlarning minimal miqdorini.
[F N h l L]	yo’q	O’lcham modifikatori. Argumentning qisqa (<i>short-h</i>) yoki uzun (<i>long-1,L</i>) ko’rinishini, adres turini aniqlaydi.
<tur belgisi>	ha	Argument qiymati almashtiriladigan tur belgisi

1.4-jadval. Format aniqlashtiruvchilari va ularning vazifasi.

Almashtiriladigan tur belgisining qabul qilishi mumkin bo’lgan belgilari quyidagi jadvalda keltirilgan:

Berilganlar turi	Kutilayotgan qiymat	Chiqish formati
Son qiymatlari		
d	Butun son	Ishorali o'nlik butun son
i	Butun son	Ishorali o'nlik butun son
o	Butun son	Ishorasiz sakkizlik butun son
u	Butun son	Ishorasiz o'nlik butun son
x	Butun son	Ishorasiz o'n oltilik butun son (a,b,c,d,e,f belgilari ishlatiladi)
X	Butun son	Ishorasiz o'n oltilik butun son (A,B,C,D,E,F belgilari ishlatiladi)
f	Suzuvchi nuqtali son	[-]dddd.dddd ko'rinishidagi suzuvchi nuqtali son
e	Suzuvchi nuqtali son	[-]d.dddd yoki e[+/-]ddd ko'rinishidagi suzuvchi nuqtali son
g	Suzuvchi nuqtali son	Ko'rsatilgan aniqlikka mos e yoki f shaklidagi suzuvchi nuqtali son
E,G	Suzuvchi nuqtali son	Ko'rsatilgan aniqlikka mos e yoki f shaklidagi suzuvchi nuqtali son. e format uchun 'E' belgisi chiqariladi.
Belgilar		
s	Satrga ko'rsatkich	o-belgisi uchramaguncha yoki ko'rsatilgan aniqlikka erishilmaguncha belgilar oqimga chiqariladi.
c	Belgi	Bitta belgi chiqariladi
%	Hech narsa	'%' belgisi oqimga chiqariladi
Ko'rsatkichlar		
n	int ko'rsatkich(int * arg)	%n argumentigacha muvaffaqiyatli chiqarilgan belgilar soni, aynan int ko'rsatkichi bo'yicha adresda saqlanadi
p	Ko'rsatkich	Argumentni YYYY:ZZZZ yoki ZZZZ ko'rinishidagi o'n oltilik songa aylantirib oqimga chiqaradi.

1.5-jadval. Print() funksiyasining almashtiriladigan tur belgilari.

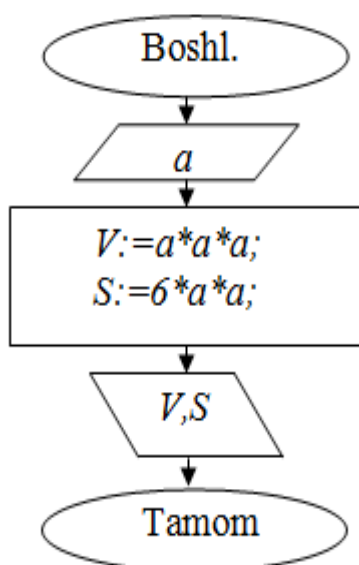
Endi berilganlar qiymatlarini oqimdan o'qish va oqimga chiqarishda *scanf()* va *printf()* funksiyalaridan foydalanishga misol keltiramiz:

```
#include <stdio.h>
int main()
{
    int son,natija;
    float hson;
    char belgi, satr[60];
    printf("\nButun va suzuvchi nuqtali sonlarni,");
    printf("\nbelgi va satrni kiriting\n");
    natija=scanf("%d %f %c %s", &son, &hson, &belgi, satr);
    printf("\nOqimdan %d ta qiymat o'qildi ",natija);
    printf("va ular quyidagilar:");
    printf("\n %d %f %c %s \n", son, hson, belgi, satr);
    return 0;
}
```

Dastur foydalanuvchidan butun, suzuvchi nuqtali son, belgi va satrni kiritishni so'raydi. Bunga javoban foydalanuvchi tomonidan

«8 17.07 F Salom» qiymatlari kiritilsa, ekranga, Oqimdan 4 ta qiymat o'qildi va ular quyidagilar: «8 17.07 F Salom» lar alohida-alohida satrlarda chop etiladi.

Masala. Kubning qirrasi *a* berilganda, uning hajmi va to'la sirtini hisoblovchi algoritim va dastur tuzilsin.



```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int a;
    /*Kub qirrasi uzunligini
    kiritish*/
    cin>>a;
    //Kubning hajmini hisoblash
    int V=a*a*a;
    /*Kubning to'la sirtini
    hisoblash*/
    int S=6*a*a;
    /*Natijalarni ekranga
    chiqarish*/
    cout<<"V="<<V<<"\t S="<<S;
    system("PAUSE");
    return EXIT_SUCCESS; }
```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 1.061 Kvadratning tomoni a berilgan bo'lsa, uning perimetri p hisoblansin.
- 1.062 Kvadratning tomoni a berilgan bo'lsa, uning yuzasi s hisoblansin.
- 1.063 To'g'ri to'rtburchakning tomonlari a va b berilgan bo'lsa, uning yuzasi s va perimetri p hisoblansin.
- 1.064 Aylana diametri d berilgan bo'lsa, uning uzunligi l hisoblansin. (Bu yerda $\pi = 3.14$ deb olinsin).
- 1.065 Kub qirrasining uzunligi a berilgan bo'lsa, kubning hajmi v va uning to'la sirti s hisoblansin.
- 1.066 To'g'ri burchakli parallelepipedning qirralari a , b , c berilgan bo'lsa, uning hajmi v va to'la sirti s lar hisoblansin.
- 1.067 Doiraning radiusi r berilgan bo'lsa, uning aylana uzunligi l va yuzasi s hisoblansin. (Bu yerda $\pi = 3.14$ deb olinsin).
- 1.068 Ikkita a va b sonlar berilgan bo'lsa, ularning o'rta arifmetigi hisoblansin.
- 1.069 Ikkita nolga teng bo'lmagan a va b sonlar berilgan bo'lsa, ularning yig'indisi, ayirmasi, ko'paytmasi va bo'linmasi hisoblansin.
- 1.070 Doira aylanasing uzunligi l berilgan bo'lsa, uning radiusi r va yuzasi s hisoblansin. (Bu yerda $\pi = 3.14$ deb olinsin).
- 1.071 Doiraning yuzasi s berilgan bo'lsa, uning diametri d va aylanasi uzunligi l hisoblansin. (Bu yerda $\pi = 3.14$ deb olinsin).
- 1.072 Sonlar o'qida berilgan X_1 va X_2 nuqtalar orasidagi masofa ($X_2 - X_1$) hisoblansin.
- 1.073 Sonlar o'qida A , B , C nuqtalar berilgan ($A < B < C$) bo'lsin, AC va BC kesmalarning uzunligi va ularning yig'indisi hisoblansin.
- 1.074 Son o'qida A , B , C nuqtalar berilgan ($A < B < C$) bo'lsa, AC va BC kesmalarning uzunliklari hamda ularning ko'paytmasi hisoblansin.
- 1.075 a va b o'zgaruvchilardagi qiymatlarni o'zaro almashtirish amalga oshirilsin.
- 1.076 a , b va c o'zgaruvchilarning qiymatlari quyidagicha almashtirilsin: $a \rightarrow b$, $b \rightarrow c$, $c \rightarrow a$.
- 1.077 a , b va c o'zgaruvchilarning qiymatlari quyidagicha almashtirilsin: $c \rightarrow b$, $b \rightarrow a$, $a \rightarrow c$.

- 1.078** α burchak ($0 < \alpha < 360^\circ$) gradusda berilgan. Unga mos r radian ($r = \frac{\pi}{180} \cdot \alpha$) hisoblansin. (Bu yerda $\pi = 3.14$ deb olinsin).
- 1.079** r burchak ($0 < r < 2\pi$) radianda berilgan. Uning gradusdagi qiymati α ($\alpha = \frac{180}{\pi} \cdot r$) hisoblansin. (Bu yerda $\pi = 3.14$ deb olinsin).
- 1.080** t harorat Ferenget birligida berilgan. Uning Sel'siy birligidagi qiymati hisoblansin. Bu yerda $t_c = (t_f - 32) \cdot \frac{5}{9}$.
- 1.081** t harorat Sel'siy birligida berilgan. U Ferenget birligiga o'tkazilsin va bu yerda $t_f = 9/5 \cdot t_c + 32$.
- 1.082** $ax + b = 0$ ($a \neq 0$) chiziqli tenglama koeffisientlari a va b lar berilgan bo'lsa, x hisoblansin.
- 1.083** Masofa s sm da berilgan. Santimetr metrda ifodalansin va natijaning butun qismi chop etilsin.
- 1.084** Og'irlik m grammda berilgan. Butunga bo'lish amalidan foydalanib, uning kg dagi ifodasi hisoblansin.
- 1.085** Fayl o'lchami baytda berilgan bo'lsa, butunga bo'lish amali yordamida uning kilobaytdagi ifodasi hisoblansin.
- 1.086** a va b butun ($a \geq b$) musbat sonlar berilgan bo'lib, ular kesmalarning uzunligini bildiradi. b kesma a kesmani to'liq qoplaydi yoki bir necha b uzunlikdagi kesmalar yig'indisi a kesмага teng deb hisoblab, a kesmaning nechta b kesmadan tashkil topishi aniqlansin.
- 1.087** a va b butun ($a \geq b$) musbat sonlar berilgan bo'lib, ular kesmalarning uzunligini bildiradi. Bir nechta b kesma uzunliklari yig'indisining mumkin bo'lgan eng kattasiga teng a kesma uzunligi ajratilsin.
- 1.088** Ikki xonali butun son berilgan. Dastlab uning chap qismidagi, so'ngra o'ng qismidagi raqami alohida-alohida qilib chop etilsin. Bu ishni bajarishda butunga bo'lish va qoldiqni hisoblash amallaridan foydalanilsin.
- 1.089** Ikki xonali butun son berilgan, uning raqamlari yig'indisi va ko'paytmasi hisoblansin.
- 1.090** Ikki xonali butun son berilgan, uning raqamlari o'rnini almashtirish natijasida hosil bo'lgan son chop etilsin.

B toifa

- 1.091** Ikkita manfiy bo'lmagan a va b sonlar berilgan bo'lsa, ularning o'rta geometrigi hisoblansin.
- 1.092** Ikkita nolga teng bo'lmagan a va b sonlar berilgan bo'lsa, ularning yig'indisi, ayirmasi, ko'paytmasi hamda bo'linmasining modullari hisoblansin.
- 1.093** To'g'ri burchakli uchburchakning a va b katetlari berilgan bo'lsa, uning gipotenuzasi c hamda perimetri p hisoblansin.
- 1.094** Radiuslari r_1 va r_2 bo'lgan ($r_1 > r_2$) hamda markazlari umumiy 2 ta doira berilgan. Birinchi doiraning yuzi s_1 , ikkinchi doiraning yuzi s_2 hamda tashqi doiraning ichida va ichki doiraning tashqarisida joylashgan yuza s_3 ($s_3 = s_1 - s_2$) hisoblansin.
- 1.095** To'g'ri burchakli to'rtburchakning qarama-qarshi uchlari (x_1, y_1) va (x_2, y_2) koordinatalar bilan berilgan. Shu to'g'ri to'rtburchakning tomonlarini koordinata o'qlariga parallel deb qaralib, uning perimetri hamda yuzasi hisoblansin.
- 1.096** Tekislikda koordinatalari bilan berilgan ikki nuqta orasidagi masofa hisoblansin.
- 1.097** Uchburchakning uchlari (x_1, y_1) , (x_2, y_2) , (x_3, y_3) koordinatalar bilan berilgan bo'lsa, ikki nuqta orasidagi masofani topish, hamda Geron formulalaridan foydalanib, shu uchburchakning perimetri va yuzasi hisoblansin.
- 1.098** x ning ($x \in \mathbb{Z}$) berilgan qiymatida $y = 3x^6 - 6x^3 - 7$ ifoda hisoblansin.
- 1.099** x ning ($x \in \mathbb{Z}$) berilgan qiymatida $y = 4(x-3)^6 - 7(x-3)^3 + 2$ ifoda hisoblansin.
- 1.100** a soni berilgan bo'lsa, yordamchi o'zgaruvchilardan foydalangan holda, 3 ta ko'paytirish amalini bajarib, a^2 , a^4 , a^5 darajalar hisoblansin.
- 1.101** a soni berilgan bo'lsa, 2 ta yordamchi o'zgaruvchidan hamda 5 ta ko'paytirish amallarini bajarib, a^2 , a^3 , a^5 , a^{10} , a^{15} darajalar hisoblansin.
- 1.102** Agar x kg konfet a so'm tursa, shunday turdagi y kg konfet qancha turishi hisoblansin.
- 1.103** x kg shokolad a so'm, y kg iris b so'm tursa. 1 kg shokolad va 1 kg iris qancha turishi va ular narxidagi farq hisoblansin.
- 1.104** Turg'un suvdagi qayiqning tezligi v km/soat hamda daryo oqimining tezligi u km/soat bo'lsin ($u < v$). Qayiq ko'lda t_1 soat, daryo oqimiga qarshi t_2 soat harakatlangan bo'lsa, uning bosib o'tgan umumiy yo'li hisoblansin.

- 1.105** Birinchi avtomobilning tezligi v_1 , ikkinchisiniki esa v_2 , ular orasidagi masofa s km. Avtomobillar bir-biridan uzoqlashayotgan bo'lsa, t vaqtdan keyingi ular orasidagi masofa hisoblansin.
- 1.106** Birinchi avtomobilning tezligi v_1 , ikkinchisiniki v_2 , ular orasidagi masofa s km bo'lib, avtomobillar bir-biriga tomon harakatlanayotgan bo'lsa, t vaqtdan keyingi ularning bosib o'tgan umumiy yo'li hisoblansin.
- 1.107** $ax^2+bx+c=0$ kvadrat tenglamaning a, b, c , ($a \neq 0$) koeffitsientlari berilgan bo'lsa, uni musbat diskriminantga ega deb hisoblab, uning ildizlari hisoblansin.
- 1.108** $a_1, b_1, c_1, a_2, b_2, c_2$ koeffitsientlar bilan berilgan, quyidagi ko'rinishga ega bo'lgan
- $$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$
- chiziqli tenglamalar sistemasining yechimi hisoblansin.
- 1.109** Uch xonali butun son berilgan bo'lsa, butunga bo'lish amalidan bir marta foydalanib uning birinchi raqami chop etilsin.
- 1.110** Uch xonali butun son berilgan bo'lsa, uning oxirgi va o'rta xonasidagi raqamlari chop etilsin.
- 1.111** Uch xonali butun son berilgan, uning raqamlari yig'indisi chop etilsin.
- 1.112** Uch xonali butun son berilgan. Uni teskari tomondan o'qiganda hosil bo'ladigan son chop etilsin.
- 1.113** Uch xonali butun son berilgan, uning birinchi raqamini olib, oxiriga o'tkazish natijasida hosil bo'ladigan son chop etilsin.
- 1.114** Uch xonali butun son berilgan. Uning o'ng tomonidagi 1-raqami olinib, chap tomonidan joylashtirilsin va hosil bo'lgan son chop etilsin.
- 1.115** Uch xonali butun son berilgan. Uning o'nlik va yuzlik xonalaridagi raqamlarini almashtirish natijasida hosil bo'ladigan son chop etilsin.
- 1.116** Uch xonali butun son berilgan. Uning o'nlik va birlik xonalaridagi raqamlarini almashtirish natijasida hosil bo'ladigan son chop etilsin.
- 1.117** 999 dan katta bo'lgan butun son berilgan. 1 marta butunga bo'lish va 1 marta qoldiqni aniqlash amallaridan foydalanib, shu sonning 100 lik xonasidagi raqam aniqlansin.
- 1.118** 9999 dan katta bo'lgan butun son berilgan. 1 marta butunga

bo‘lish va 1 marta qoldiqni aniqlash amallaridan foydalanib, shu sonning 1000 lik xonasidagi raqam aniqlansin.

1.119 Sutkaning n -sekundi berilgan bo‘lsa, sutka boshidan buyon necha minut o‘tganligi aniqlansin.

1.120 Sutkaning n -sekundi berilgan bo‘lsa, sutka boshidan buyon necha soat o‘tganligi aniqlansin.

1.3. Mantiqiy ifodalar

Mantiqiy tur – mantiqiy ifodalarning qiymatlarini qabul qiluvchi o‘zgaruvchilarni e‘lon qilishda ishlatiladi. Bu turga tegishli o‘zgaruvchilar 0 (*yolg‘on*) yoki 1 (*rost*) qiymatlardan birini qabul qiladi.

Dasturlash tillarida tuzilgan mantiqiy ifodalar to‘liq matematik mantiq qonuniyatlariga asoslanadi.

Matematik mantiq – fikrlashning shakli va qonuniyatlari haqidagi fan. Uning asosini mulohazalar hisobi tashkil qiladi. *Mulohaza* – bu ixtiyoriy jumla bo‘lib, unga nisbatan rost yoki yolg‘on fikrni bildirish mumkin. *Masalan* « $-5 > -25$ », « 11 –juft son», «*Toshkent-Hindistonning poytaxti*», «*Samarqand–tarixiy shahar*» va hokazolar. Lekin « $78\ 987\ 256$ katta son» jumlasini to‘g‘ri mulohaza hisoblanmaydi, chunki «katta son» tushunchasi juda ham nisbiy, ya‘ni katta son deganda qanday sonni nazarda tutish aniq emas. Shuning uchun ham yuqoridagi jumlaning rost yoki yolg‘onligi haqida fikr bildirish qiyin.

Mulohazalarning *rost* yoki *yolg‘on* bo‘lishi holatlarga bog‘liq ravishda ham o‘zgarishi mumkin: *Masalan*, «*hozir soat 13.00*» jumlasining rost yoki yolg‘onligi qaralayotgan vaqtga bog‘liq. Xuddi shunday « $x > 0$ » ifodaning qiymati x o‘zgaruvchisining qiymatiga bevosita bog‘liq bo‘ladi.

C oilasidagi dasturlash tillarida mantiqiy ifodalar qurish uchun quyidagi taqqoslash amallari kiritilgan “ $>$ ” (*katta*), “ $<$ ” (*kichik*), “ $>=$ ” (*katta yoki teng*), “ $<=$ ” (*kichik yoki teng*), “ $==$ ” (*teng*), “ $!=$ ” (*teng emas*).

C++ tilida mantiqiy tur «Bul algebrasi» asoschisi, Angliyalik matematik Jorj Bul sharafiga *bool* kalit so‘zi bilan ifodalangan.

“*Bul algebrasi*” da mantiqiy mulohazalar ustida uchta amal aniqlangan:

Inkor – A mulohazaning inkori deganda A rost bo‘lganda yolg‘on, yolg‘on bo‘lganda rost qiymat qabul qiluvchi mulohazaga aytiladi. C++

tilida inkor – ‘!’ belgisi bilan ifodalanadi. *Masalan*, A mulohazaning inkori «! A » ko‘rinishida yoziladi;

Konyunksiya – ikkita A va B mulohazalar konyuksiyasi yoki mantiqiy ko‘paytmasi « $A \&\& B$ » ko‘rinishida yoziladi. Bu mulohaza qiymati A va B mulohazalarning har ikkalasi rost bo‘lgandagina rost bo‘ladi, aks holda esa yolg‘on bo‘ladi (« $\&\&$ » amali «va» deb o‘qiladi). *Masalan*, «bugun oyning 8-kuni va bugun payshanba» mulohazasi oyning 8-kuni payshanba bo‘lgan kunlar uchungina rost bo‘ladi;

Dizyunksiya – ikkita A va B mulohazalar dizyunksiyasi yoki mantiqiy yig‘indisi « $A || B$ » ko‘rinishida yoziladi. Bu mulohaza rost bo‘lishi uchun A yoki B mulohazalardan hech bo‘lmaganda bittasi rost bo‘lishi yetarli. (« $||$ » amali «yoki» deb o‘qiladi). *Masalan*, bugun “dushanba yoki chorshanba” mulohazasi haftaning dushanda yoki chorshanba kunlari uchun rost hisoblanadi;

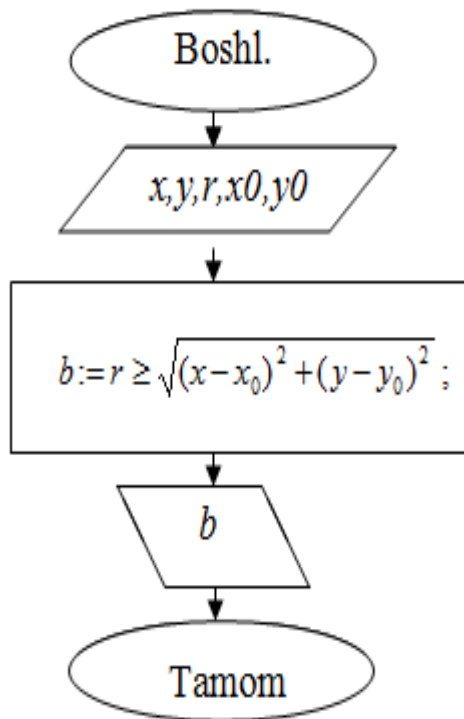
Yuqorida keltirilgan fikrlar asosida mantiqiy amallar uchun rostlik jadvali aniqlangan:

Mulohazalar		Mulohazalar ustida amallar		
A	B	$!A$	$A\&\&B$	$A B$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

1.6-jadval. Rostlik jadvali.

Mantiqiy tur qiymatlari ustida mantiqiy ko‘paytirish, qo‘shish va inkor amallarini qo‘llash orqali murakkab mantiqiy ifodalarni qurish mumkin. Misol uchun, « x –manfiy va y ning qiymati $[-1..3]$ sonlar oralig‘iga tegishli emas» mulohazasining mantiqiy ifodasi quyidagicha bo‘ladi: $(x < 0) \&\& (y < -1) || (y > 3)$.

Masala. Tekislikda markazining koordinatalari x, y , radiusi r bo‘lgan aylana, hamda nuqta x_0, y_0 koordinatalari bilan berilgan. Berilgan nuqtaning aylanaga tegishli ekanligini aniqlovchi algoritmi va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
#include <math.h>
using namespace std;
int main(int argc, char
*argv[])
{ float x,y,r,x0,y0;
/*Aylana markazining
koordinatalari va radiusini
kiritish*/
    cin>>x>>y>>r;
    /*Berilgan nuqta
koordinatalarini      kiritish*/
    cin>>x0>>y0;
    /*Berilgan nuqtaning aylana
ichida yotishlik shartini
tekshirish*/
    bool b=r>=sqrt(pow(x-x0,2)
+pow(y-y0,2));
    //Natijani chiqarish
    cout<<"b="<<b;
    system("PAUSE");
    return EXIT_SUCCESS;}
    
```

Berilgan masalalar uchun mantiqiy mulohazalar bajarilganda 1 (rost), aks holda 0 (yolg'on) qiymat qaytaruvchi algoritmlar va dasturlar tuzilsin:

A toifa

- 1.121 $0 < x < 1$;
- 1.122 $x = \max(x, y, z)$;
- 1.123 $x \neq \max(x, y, z)$ (inkor amalidan foydalanilmasin);
- 1.124 a va b mantiqiy o'zgaruvchilardan hech bo'lmaganda bittasi 1 (rost) qiymatga ega;
- 1.125 a va b mantiqiy o'zgaruvchilardan hech bo'lmaganda bittasi 0 (yolg'on) qiymatga ega;
- 1.126 Har ikkala a va b mantiqiy o'zgaruvchilarning qiymatlari bir vaqtda 1 (rost) qiymatga ega;
- 1.127 $x \in [0; 1]$;
- 1.128 $x \notin [0; 1]$;
- 1.129 $x \in [2; 5]$ yoki $x \in [-1; 1]$;

- 1.130 $x \notin [2;5]$ yoki $x \notin [-1;1]$;
- 1.131 x, y, z sonlarining har biri musbat;
- 1.132 x, y, z sonlarining hech bo'lmaganda biri musbat;
- 1.133 x, y, z sonlarining hech biri musbat emas;
- 1.134 x, y, z sonlarining faqat bittasi musbat;
- 1.135 Mantiqiy o'zgaruvchi a ning qiymati 1 (rost), b ning qiymati esa 0 (yolg'on) bo'lgan hol uchun;
- 1.136 y kabisa yili bo'lsa (kabisa yili 4 ga karrali yillar hisoblanadi. Biroq, 100 ga karrali yillar orasida faqat 400 ga karralisi kabisa yili deyiladi. Masalan, 1700, 1800, 1900 oddiy yillar, 2000 esa kabisa yilidir).
- 1.137 n va k natural sonlari bir xil juftlikka ega;
- 1.138 n va k natural sonlari bir xil juftlikka ega emas;
- 1.139 n va k butun sonlari bir xil ishorali;
- 1.140 n va k butun sonlari turli ishorali;
- 1.141 a, b mantiqiy o'zgaruvchilardan faqat bittasi 1 (rost) qiymatga ega;
- 1.142 a, b mantiqiy o'zgaruvchilardan faqat bittasi 0 (yolg'on) qiymatga ega;
- 1.143 a, b, c mantiqiy o'zgaruvchilardan faqat bittasi 1 (rost) qiymatga ega bo'lsa;
- 1.144 a, b, c mantiqiy o'zgaruvchilardan bir vaqtda ikkitasi 1 (rost) qiymatga ega bo'lsa;
- 1.145 a, b, c mantiqiy o'zgaruvchilardan faqat bittasi 0 (yolg'on) qiymatga ega bo'lsa;
- 1.146 a, b, c mantiqiy o'zgaruvchilardan faqat ikkitasi 0 (yolg'on) qiymatga ega bo'lsa;
- 1.147 a butun son berilgan bo'lsa, u musbatlikka tekshirilsin.
- 1.148 a butun son berilgan bo'lsa, u juftlikka tekshirilsin.
- 1.149 a butun son berilgan bo'lsa, u toqlikka tekshirilsin.
- 1.150 a va b butun sonlar berilgan bo'lsa, ularning $2 \leq a \leq b \leq 5$ shartni qanoatlantirishi tekshirilsin.

B toifa

- 1.151 a, b, c butun sonlar berilgan bo'lsa, ularning $a \leq b \leq c$ shartni qanoatlantirishi aniqlansin.
- 1.152 a, b, c butun sonlar berilgan. b sonining, a va c sonlar orasida yotishi aniqlansin.
- 1.153 a va b butun sonlar berilgan. Ularning bir vaqtda toq

- bo'lmashligi aniqlansin.
- 1.154** a va b butun sonlar berilgan bo'lsa, ularning hech bo'lmaganda bittasi toq ekanligi aniqlansin.
- 1.155** a va b butun sonlar berilgan bo'lsa, bu sonlardan biri toq ekanligi aniqlansin.
- 1.156** a va b butun sonlar berilgan bo'lsa, ularning bir xil juftlikka ega ekanligi aniqlansin.
- 1.157** a , b , c butun sonlar berilgan. Ularning har biri musbat ekanligi tekshirilsin.
- 1.158** a , b , c butun sonlar berilgan bo'lsa, ularning hech bo'lmaganda bittasi musbat ekanligi aniqlansin.
- 1.159** a , b , c butun sonlar berilgan bo'lsa, ulardan faqat bittasi musbat bo'lishi aniqlansin.
- 1.160** a , b , c butun sonlar berilgan bo'lsa, ulardan faqat ikkitasi bir vaqtda musbat ekanligi aniqlansin.
- 1.161** Butun musbat son berilgan bo'lsa, uning bir vaqtda juft va ikki xonali ekanligi aniqlansin.
- 1.162** Butun musbat son berilgan bo'lsa, uning bir vaqtda toqligi va uch xonali ekanligi aniqlansin.
- 1.163** a , b , c butun sonlar berilgan. b sonining, a va c sonlar orasida yotishi aniqlansin.
- 1.164** a va b butun sonlar berilgan. Ularning bir vaqtda toq bo'lmashligi aniqlansin.
- 1.165** a va b butun sonlar berilgan bo'lsa, ularning hech bo'lmaganda bittasi toq ekanligi aniqlansin.
- 1.166** Berilgan uchta sondan juftliklar hosil qilingan. Shu juftliklarning hech bo'lmaganda bittasidagi sonlar o'zaro teng bo'lishi aniqlansin.
- 1.167** Berilgan uchta butun sonlar orasidan olingan juftliklardan hech bo'lmaganda bittasidagi sonlar ishoralari bilan farq qilishi aniqlansin.
- 1.168** Uch xonali son berilgan. Bu son raqamlarining har xil ekanligi aniqlansin.
- 1.169** Uch xonali son berilgan. Uning raqamlari o'suvchi ketma-ketlik tashkil etishi aniqlansin.
- 1.170** Uch xonali son berilgan. Uning raqamlari o'suvchi yoki kamayuvchi ketma-ketlik tashkil etishi aniqlansin.
- 1.171** To'rt xonali son berilgan. Uni chapdan o'ngga va o'ngdan chapga o'qiganda bir xil o'qilishi aniqlansin.

- 1.172 a, b, c sonlar berilgan ($a \neq 0$). Bu sonlarni kvadrat tenglama koeffisientlari deb hisoblab, shu kvadrat tenglamaning haqiqiy yechimga ega ekanligi aniqlansin.
- 1.173 x, y sonlari berilgan. Ularni tekislikdagi nuqta koordinatalari deb hisoblab, shu nuqtaning 2-chorakda yotishi aniqlansin.
- 1.174 x, y sonlari berilgan. Ularni tekislikdagi nuqta koordinatalari deb hisoblab, ularning 2- yoki 3-chorakda yotishi aniqlansin.
- 1.175 Tekislikda nuqta x va y koordinatalari bilan berilgan. Shu nuqta (yuqori chap burchagi (x_1, y_1) , quyi o'ng burchagi (x_3, y_3) bo'lgan, hamda tomonlari koordinata o'qlariga parallel) to'g'ri burchakli to'rtburchakning ichida yotishi yoki yotmasligi aniqlansin.
- 1.176 Uchta butun son berilgan bo'lsa, shu sonlarning uchburchakning tomonlarini tashkil etishi aniqlansin.
- 1.177 a, b, c butun sonlar berilgan bo'lib, ular uchburchakning tomonlarini tashkil etsa, shu uchburchakning teng yonli ekanligi aniqlansin.
- 1.178 a, b, c butun sonlar berilgan bo'lib, ular uchburchakning tomonlarini tashkil etsa, shu uchburchakning to'g'ri burchakli ekanligi aniqlansin.
- 1.179 Shaxmat taxtasining quyi chap burchagini koordinata boshi deb hisoblab, (uning kataklarini 1 dan 8 gacha butun sonlar bilan belgilangan), shu burchakdagi katakcha qora bo'lganda, berilgan katak qora bo'lsa rost, aks holda yolg'on ekanligi aniqlansin.
- 1.180 Shaxmat taxtasining quyi chap burchagini koordinata boshi deb hisoblab, (uning kataklari 1 dan 8 gacha butun sonlar bilan belgilangan), uning ikkala qismidan bittadan katakcha berilganda, shu katakchalar bir xil rangdaligi aniqlansin.

1.4. Bobga doir qiyinroq masalalar

1.181

5 karra 5 [10].

Ali va Vali maktabda o'qishadi. Yaqinda Vali Aliga 5 raqami bilan tugaydigan natural sonlarni kvadratga ko'tarish usulini o'rgatdi. Endi Ali 5 raqami bilan tugaydigan ikki xonali (ba'zi bir uch xonali) sonlarni osonlik bilan kvadratga ko'tarishi mumkin. Bu usul quyidagicha amalga oshiriladi: 5 raqami bilan tugaydigan sonlarni kvadratga ko'tarish uchun berilgan sonning oxirgi raqamini o'chirilib, qolgan son o'zidan birga ortiq bo'lgan songa

ko'paytiriladi va chiqqan son oxiriga (o'ng tomoniga) 25 soni yozib qo'yiladi.

Masalan, 115 sonini kvadratga ko'paytirish uchun, shunchaki 11 ni 12 ga ko'paytirib, ko'paytma oxiriga 25 sonini yozib qo'yamiz, ya'ni $11 \cdot 12 = 132$ oxiriga 25, javob 13225. Demak 115 ning kvadrati 13225 ga teng. Alining o'rganganlarini sinab ko'rishi uchun 5 raqami bilan tugaydigan sonlarni yuqoridagi usul yordamida kvadratga ko'taradigan dastur yozilsin.

Berilganlar:

Yagona qatorda 5 raqami bilan tugaydigan, $1 \cdot 10^3$ dan oshmaydigan, bitta natural son yoziladi.

Natija:

Berilgan natural sonning kvadrati chiqariladi.

<i>Berilganlar:</i>	<i>Natija:</i>
15	225
75	5625
4255	18105025
0	0

1.182

Sветофор [11].

Ali chorrahaga keldi va shu vaqtda svetofor qizil yonganligini ko'rdi. Ali qiziqib qoldi, svetofor chiroqlari qanday aniqlik bilan yonmoqda:

qizil –sariq – yashil –sariq –qizil – sariq –yashil - ...

Navbatdagi yashil chiroq yonganda Ali yo'lni kesib o'tishga ahd qildi. Bu vaqtda yashil chiroq i - marta yongandi. Shu vaqtgacha necha marta qizil chiroq (Ali kelgan vaqtni hisobga olgan holda) va necha marta sariq chiroq yonganligini aniqlovchi dastur tuzing.

Berilganlar:

Svetoforning yashil chirog'i necha marta yonganligini ko'rsatuvchi bitta i ($1 \leq i \leq 100$) soni berilgan.

Natija:

Necha marta qizil chiroq va necha marta sariq chiroq yonganligini bildiruvchi ikkita son chiqarilsin.

Berilganlar:	Natija:
2	2 3

1.183

Kvadratlar yig'indisi.

Ikki a va b sonlari berilgan ($-10^9 \leq a, b \leq 10^9$). $a^2 + b^2$ berilgan sonlarning yig'indisini chiqaruvchi dastur tuzilsin. Yordam: Borland Delphi va Free Pascal uchun `int64`, Visual C/C++ uchun `__int64`, GNU C/C++ uchun `long long` yoki Java uchun `long` dan foydalaning.

Berilganlar: a va b sonlari berilgan.*Natija:*

Ularning kvadratlar yig'indisi topilsin.

Berilganlar:	Natija:
2 3	13
17 -18	613

1.184

Doston boshqotirmasi [12].

Doston ayni damda dasturlash bo'yicha olimpiada masalalarini yechish bilan band. Chunki u yaqinda bo'lib o'tadigan olimpiadada qatnashishi kerak. Doston bir masala haqida bosh qotirmoqda. U uzoq fikrlab masalaning sharti berilgan sonning o'n birga bo'linish yoki bo'linmasligini tekshirishdan iborat ekanligini aniqladi. Unga bu masalani yechishda yordam bera olasizmi?

*Berilganlar:*Bitta butun son- N ($1 \leq N \leq 10^{10000}$) kiritiladi.*Natija:*

“Yes”, agar kiritilgan son o'n birga bo'linsa, aks holda “No” so'zini chiqarilsin.

Berilganlar:	Natija:
11	Yes
111	No

1.185

Suv o'lchagich.

Quruvchilarga yangi bino qurish topshirildi. Ular bino quradigan

yerga suv bormaydi. Qurilish esa albatta suvsiz bo'lmaydi. Bu masalani hal etish uchun qurilish firmasi vaqtida suv yetkazib berishi kerak, aks holda ish to'xtab qoladi. Suv idishi silindrsimon bo'lib (yotgan holatda) uning uzunligi L , radiusi R . Idishda qancha suv borligini aniqlovchi maxsus uzun tayoqcha bo'lib, uni idishning tepasidan idishning pastiga tekkunicha solamiz va tayoqchaning qancha uzunligi ho'l bo'lsa, shunga qarab idishda qancha suv borligi aniqlanadi. Tayoqchaning ho'l bo'lgan uzunligi H ga teng. Suv tugashidan avvalroq ular suv olib kelishlari haqida xabar berishlari kerak. Quruvchilar sizdan idishda qancha suv borligini hisoblovchi dastur tuzishni iltimos qilishmoqda.

Berilganlar:

Bitta satrda uchta haqiqiy L , R va H sonlari bo'sh joylar orqali beriladi ($0 < L, R \leq 100$, $0 \leq H \leq 2 \cdot R$).

Natija:

Idishda qancha litr suv borligi 10^{-3} aniqlikda hisoblansin.

<i>Berilganlar:</i>	<i>Natija:</i>
1 1 2	3.142
1 1 1	1.571
2 1 0.97	3.022

1.186

Tejamkor robot.

Fazoviy kema qanotlaridagi quyoshdan quvvat oluvchi moslamalarni robot muntazam ravishda tozalab turadi. Moslama tomonlari a va b o'lchamga ega to'g'ri to'rtburchak shaklidagi n ta plastinkadan iborat. Robot 1 birlik yuzani tozalash uchun k gr tozalovchi suyuqlik sarflasa, moslamani to'liq tozalash uchun sarflanadigan suyuqlik miqdori hisoblansin.

Berilganlar:

Yagona satrda bo'sh joylar bilan ajratilgan n , a , b , k sonlari kiritiladi.

Natija:

Sarflanadigan tozalovchi suyuqlik miqdorini bildiradigan yagona son chiqariladi.

<i>Berilganlar:</i>	<i>Natija:</i>
2 2 2 2	16
4 3 2 1	24

1.187

Budilnik[12].

Yo'ldosh juda mas'uliyatli talaba. U hamisha ertalab o'qishga, o'z vaqtida borishni yoqtiradi. Ertalab uyg'onish barcha talabalar uchun eng muommoli ish hisoblanadi, shu qatorda Yo'ldosh uchun ham. Shuning uchun u har tong uygo'nish uchun budilnikdan foydalanadi. Budilnikni ertalabki ma'lum vaqtga qo'yadi. Budilnik qo'yilgan vaqtga kelganda chalishni boshlaydi va 1 daqiqa davomida chalib turadi. Birinchi marta chalgandan uyg'onish qiyin, shuning uchun budilniklarda takrorlanish davri ham bo'ladi. Agar shu bir daqiqa ichida uyg'onmasa ma'lum vaqt o'tgandan so'ng budilnik yana chalishni boshlaydi va hokozo. Masalan agar budilnik 05:00 ga qo'yilgan va takrorlanish vaqti 3 minut bo'lsa, budilnik 05:01(05:01 vaqt momentining o'zi tegishli emas)da chalishdan to'xtaydi va keyingi chalish 05:04 da boshlanadi. Yo'ldosh uxlashdan oldin budilnik vaqti va takror chalish vaqtini belgilaydi. U bir necha marta budilnik chalishdan so'ng uyg'onadi va soatga qaraydi. Uni faqat bitta savol qiziqtiradi. Budilnik nechanchi marta chalayotgan ekan? Bu savolga javob berishda unga yordam bering.

Berilganlar:

Birinchi qatorda budilnik qaysi vaqtga qo'yilgani, ikkinchi qatorda esa Yo'ldoshning uyg'ongan vaqti $hh:mm$ ($05 \leq hh \leq 12$, $00 \leq mm \leq 59$) formatda beriladi. Bu vaqtlar 05:00 dan 12:00 gacha bo'lishi mumkin va ikkinchi vaqt hamisha birinchi vaqtdan kam bo'lmaydi va bu ikkala vaqt bitta sutka vaqtlari. Uyg'onish vaqtida budilnik chalib turganligi kafolatlanadi. Uchinchi qatorda 1 dan 20 gacha bo'lgan bitta butun son – budilnikning takror chalishgacha ketadigan vaqt minutda beriladi.

Natija:

Bitta butun sonni – Yo'ldoshning uyg'ongan vaqtda budilnik nechanchi marta chalayotganligi chiqarilsin.

<i>N_o</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	05:00 05:00 3	1
2	05:00 10:12 1	157

1.188

Uchrashuv [12].

To'rtta do'st G'iyos, Jamshid, Shavkat va Yo'ldosh ko'p ishlarni birgalikda qilishadi. Endi ular o'zlariga uy qurishmoqchi. Har bir uyni birgalikda qurishayabdi. G'iyosning uyi (x_1, y_1) nuqtaga, Jamshidning uyi (x_2, y_2) nuqtaga, Shavkatning uyi (x_3, y_3) nuqtaga qurildi. Bu nuqtalar bitta chiziqda yotmaydi. Endi Yo'ldoshning uyini qurishmoqchi. Ular har kecha kimningdir uyida uchrashib, birgalikda masala yechishadi va karta o'ynashadi. Yo'ldosh juda shnavanda yigit bo'lganligi uchun har kecha uning uyida uchrashishni rejalashtirishdi. Endi uning uyini shunday nuqtaga qurishlari kerakki, bu nuqtadan G'iyos, Jamshid va Shavkatning uylarigacha bo'lgan masofalar teng bo'lishi kerak. Ular kuchli dasturchi bo'lganliklari uchun bu nuqtani dastur tuzib topishga qaror qilishdi. Siz ham bu nuqtani topishga harakat qilib ko'ring, balki ulardan oldinroq toparsiz.

Berilganlar:

Birinchi qatorda $x_1, y_1, x_2, y_2, x_3, y_3$ sonlari berilgan. Ularning barchasi butun va modul jihatdan 10^4 dan oshmaydi. Bu uch nuqta bir tog'ri chiziqda yotmaydi.

Natija:

Yo'ldoshning qurilishi kerak bo'lgan uyining x va y koordinatalarni 10^{-6} aniqlikda bitta bo'sh joy bilan ajratib chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
1 1 4 2 2 5	2.045455 2.863636

II Bob. Tarmoqlanuvchi algoritmlar va ularni dasturlash

2.1. Tarmoqlanish va shartsiz o'tish jarayonlari

Ko'pgina masalalar yuzaga keladigan turli holatlarga bog'liq ravishda qaror qabul qilishni talab etadi. Bu borada C++ tili dasturning alohida bo'laklarining bajarilish tartibini boshqarishga imkon beruvchi qurilmalarning yetarlicha katta majmuasiga ega. Masalan, dastur bajarilishining birorta qadamida qandaydir shartni tekshirish natijasiga ko'ra boshqaruvni dasturning u yoki bu bo'lagiga uzatish mumkin (tarmoqlanuvchi algoritm). Bunday tarmoqlanishni amalga oshirish uchun shart operatorlaridan foydalaniladi.

if operatori: *if* operatori qandaydir shartni rostlikka tekshirshi natijasiga ko'ra dasturda tarmoqlanishni amalga oshiradi:

if (<shart>)<operator>;

Bu yerda <shart> mantiqiy turga tegishli o'zgaruvchi yoki mantiqiy ifoda bo'lishi mumkin.

Agar shart 0 qiymatidan farqli, ya'ni *rost* bo'lsa, <operator> bajariladi, aks holda, ya'ni shartning qiymati 0(*yolg'on*) bo'lsa, hech qanday amal bajarilmaydi va boshqaruv *if* operatoridan keyingi operatorga o'tadi (*agar u mavjud bo'lsa*).

C++ tilining qurilmalari operatorlarni blok ko'rinishida tashkil qilishga imkon beradi. Blok – '{' va '}' belgilar oralig'iga olingan operatorlar ketma-ketligi bo'lib, u kompilyator tomonidan yaxlit bir operator deb qabul qilinadi. Blok ichida o'zgaruvchilar e'lonlari ham bo'lishi mumkin va blokda e'lon qilingan o'zgaruvchilar faqat shu blok ichida (ko'rinadi) amal qiladi, blokdan tashqarida (ko'rinmaydi) amal qilmaydi. Blokdan keyin ';' belgisi qo'yilmasligi mumkin, lekin blok ichidagi har bir ifoda ';' belgisi bilan yakunlanishi shart.

If – else operatori: Shart operatorining *if – else* ko'rinishi quyidagicha:

if (<shart>) <operator1>; *else* <operator2>;

Bu yerda <shart> 0 qiymatidan farqli, ya'ni *rost* bo'lsa <operator1>, aks holda <operator2> bajariladi. *If-else* shart operator mazmuniga ko'ra algoritmning tarmoqlanuvchi blokini ifodalaydi.

?: shart amali: Agar tekshirilayotgan shart nisbatan sodda bo'lsa, shart amalining '?' ko'rinishini ishlatish mumkin:

<shart> ? <ifoda1> : <ifoda2>;

Ushbu shart amali *if* shart operatoriga o‘xshash ko‘rinishda ishlaydi: agar *<shart>* 0 qiymatidan farqli, ya’ni *rost* bo‘lsa, *<ifoda1>*, aks holda *<ifoda2>* bajariladi. Odatda ifodalarning qiymatlari biror o‘zgaruvchiga o‘zlashtiriladi.

Goto operatori va nishonlar: Nishon – bu davomida ikkita nuqta (‘:’) qo‘yilgan identifikator. Nishon bilan qandaydir operator belgilanadi va keyinchalik dasturning boshqa bir qismidan unga shartsiz o‘tish amalga oshiriladi. Nishon bilan har qanday operator belgilanishi mumkin, shu jumladan e’lon operatori va bo‘sh operatorlar ham. Nishon faqat funksiyalar ichida amal qiladi.

Nishonga shartsiz o‘tish goto operatori yordamida bajariladi. Goto operatori orqali faqat uning o‘zi joylashgan funksiya ichidagi operatorlarga o‘tish mumkin. Goto operatorining sintaksisi quyidagicha:

goto <nishon>;

Ayrim hollarda, goto operatorining «sakrab o‘tishi» hisobiga xatoliklar yuzaga kelishi mumkin. *Masalan*,

```
int i=0;  
i++; if(i) goto m;  
int j;  
m: j+=i;
```

operatorlarining bajarilishi xatolikka olib keladi, chunki *j* o‘zgaruvchisining e’loni chetlab o‘tilgan.

Shartsiz o‘tish operatori dastur tuzishdagi kuchli va shu bilan birga xavfli vositalardan biri hisoblanadi. Kuchliligi shundaki, u yordamida algoritmnining «*boshi berk*» joylaridan chiqib ketish mumkin. Ikkinchi tomondan, bloklarning ichiga o‘tish, *masalan*, takrorlash operatorlarining ichiga «*sakrab*» kirish kutilmagan holatlarni yuzaga keltirishi mumkin. Shu sababli, imkon qadar goto operatoridan foydalanmaslik kerak, foydalanilgan taqdirda ham quyidagi qoidalarga amal qilish zarur – blok ichiga, *if*, *if–else* va *switch* operatorlari ichiga, hamda takrorlash operatorlari tanasiga tashqaridan “sakrab” kirish mumkin emas. Garchi, nishon yordamida dasturning ixtiyoriy joyiga o‘tish mumkin bo‘lsa ham, boshlang‘ich qiymat berish e’lonlaridan sakrab o‘tish man etiladi, lekin bloklardan sakrab o‘tish mumkin.

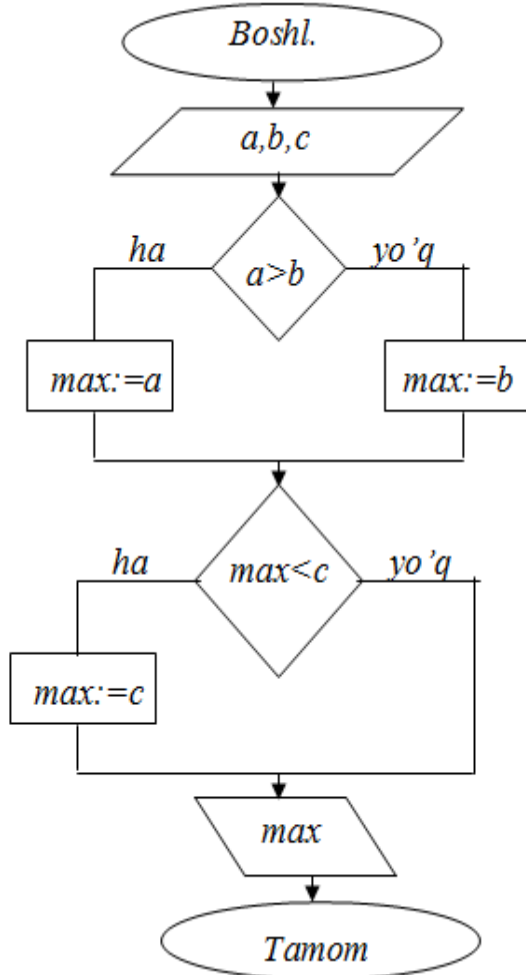
Masalan:

```
...  
goto B3A; \\ xato  
float x=0.5;  
goto B3A; \\ to‘g‘ri
```

```
{int n=10; x=n*x+8;}
B3A: cout<<"x="<<x;
```

...

Masala. Berilgan 3 ta a, b, c o'zgaruvchilardagi sonli qiymatlarning eng kattasini aniqlovchi algoritm va dastur tuzilsin.



```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char
*argv[])
{ int a,b,c,max;
/*Berilgan 3 ta
o'zgaruvchiga sonli
qiymatlar kiritish*/
cin>>a>>b>>c;
/*Dastlabki 2 ta sonning
kattasini topib, max
o'zgaruvchisiga qiymatlash*/
if (a>b) max=a;
else max=b;
/*max o'zgaruvchisidagi
qiymat bilan 3-sonni
taqqoslash*/
if (max<c) max=c;
//Natijani chiqarish;
cout<<"max="<<max;
system("PAUSE");
return EXIT_SUCCESS;}
```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 2.001 Butun son berilgan. Agar u musbat bo'lsa unga 1 qo'shilsin, aks holda o'zgarishsiz qoldirilsin. Olingan son chiqarilsin.
- 2.002 Butun son berilgan. Agar u manfiy bo'lsa unga 1 qo'shilsin, aks holda 2 ayirib tashlansin. Olingan son chiqarilsin.
- 2.003 Butun son berilgan. Agar u manfiy bo'lsa 2 ayirilsin, 0 ga teng bo'lsa 10 bilan almashtirilsin. Olingan son chiqarilsin.
- 2.004 Ikkita son berilgan. Ulardan kattasi chiqarilsin.
- 2.005 Ikkita son berilgan. Ulardan kichigining tartib raqami chiqarilsin.
- 2.006 Ikkita son berilgan. Ulardan dastlab kattasi so'ngra kichigi

navbat bilan chiqarilsin.

- 2.007** Ikkita haqiqiy turga tegishli a va b o'zgaruvchilari berilgan. Ularning qiymatlari quyidagicha qayta taqsimlansin: a ga kichigi b ga kattasi, a va b larning yangi qiymatlari chiqarilsin.
- 2.008** Ikkita butun turga tegishli a va b o'zgaruvchilar berilgan. Agar ularning qiymatlari teng bo'lmasa har bir o'zgaruvchiga qiymatlar yig'indisi berilsin, aks holda har bir o'zgaruvchiga 0 qiymatlansin. O'zgaruvchilarning natijaviy qiymatlari chiqarilsin.
- 2.009** Ikkita butun turga tegishli a va b o'zgaruvchilar berilgan. Agar ularning qiymatlari teng bo'lmasa har ikkala o'zgaruvchiga ham qiymatlarning kattasi qiymatlansin, aks holda har ikkala o'zgaruvchiga 0 qiymatlansin. O'zgaruvchilarning natijaviy qiymatlari chiqarilsin.
- 2.010** Sonlar o'qida uchta A , B , C nuqta joylashgan. B va C nuqtalarning A nuqtaga eng yaqini hamda A nuqta bilan yaqin nuqta orasidagi masofa aniqlanib chiqarilsin.
- 2.011** To'g'ri to'rtburchakning 3 ta uchi butun sonlardan iborat koordinatalar bilan berilgan. Shu uchlar orasidagi tomonlar koordinata o'qlariga parallel bo'lsa, to'rtburchakning to'rtinchi uchining koordinatasi topilsin.
- 2.012** Berilgan haqiqiy x o'zgaruvchining qiymatiga mos keluvchi f funksiyaning qiymati hisoblansin.
- $$f(x) = \begin{cases} 2\sin x, & x > 0 \\ 6 - x, & x \leq 0 \end{cases}$$
- 2.013** Butun turga tegishli x o'zgaruvchi berilgan. Uning qiymatiga mos keluvchi f funksiyaning butun qiymati hisoblansin.
- $$f(x) = \begin{cases} 2x, & x < -2 \text{ yoki } x > 2 \\ -3x, & \text{aksholda} \end{cases}$$
- 2.014** Yil nomeri berilgan. Bu yildagi kunlar soni aniqlansin (ma'lumki kabisa bo'lmagan yil 365 kundan, kabisa yili 366 kundan iborat).
- 2.015** Agar x va y sonlar teng bo'lsa, b mantiqiy o'zgaruvchiga 1 (rost), aks holda 0 (yolg'on) qiymati berilsin.
- 2.016** Berilgan a uchun $f(x)=0$ tenglamaning ildizini toping, bu yerda

$$f(x) = \begin{cases} 2ax + |a - 1|, & \text{agar } a > 0 \\ \frac{e^x}{\sqrt{1+a^2}} - 1, & \text{aks holda} \end{cases}$$

B toifa

- 2.017 Uchta butun son berilgan. Ular orasidan musbatlari soni topilsin.
- 2.018 Uchta butun son berilgan. Ular orasidan musbatlari va manfiylari soni topilsin.
- 2.019 Uchta son berilgan. Ular orasidan eng kichigi topilsin.
- 2.020 Uchta son berilgan. Ular orasidan o'rtachasi topilsin.
- 2.021 Uchta son berilgan. Ular orasidan dastlab eng kichigi so'ngra eng kattasi chiqarilsin.
- 2.022 Uchta son berilgan. Ularning ikkita kattasining yig'indisi chiqarilsin.
- 2.023 Haqiqiy turga tegishli uchta a , b , c o'zgaruvchilar berilgan. Agar o'zgaruvchilarning qiymatlari o'sish tartibida joylashgan bo'lsa, ularning qiymatlari ikki marta oshirilsin, aks holda har bir o'zgaruvchining qiymati teskarisi bilan almashtirilsin.
- 2.024 Haqiqiy turga tegishli uchta a , b , c o'zgaruvchilar berilgan. Agar o'zgaruvchilarning qiymatlari o'sish yoki kamayish tartibida joylashgan bo'lsa, har birining qiymati ikki martaga oshirilsin, aks holda o'zgaruvchilarning qiymatlari qarama-qarshisi bilan almashtirilsin. O'zgaruvchilarning natijaviy qiymatlari chiqarilsin.
- 2.025 Uchta butun son berilgan. Ulardan bittasi qolgan ikkitasidan ishoralari bilan farq qilsa, shu farq qiluvchi sonning tartib nomeri aniqlansin.
- 2.026 To'rtta butun son berilgan. Ulardan bittasi qolgan uchtasidan farq qilsa (juft toqligi bilan) bu sonning tartib nomeri chiqarilsin.
- 2.027 Tekislikda butun sonlardan iborat koordinataga ega nuqta joylashgan. Agar u koordinata boshi bilan ustma-ust tushsa 0, Ox o'qida joylashgan bo'lsa 1, Oy o'qida joylashgan bo'lsa 2, aks holda 3, qiymat chiqarilsin.
- 2.028 Ox va Oy o'qlarida yotmaydigan nuqta koordinatalari berilgan. Uning qaysi chorakka tegishli ekanligi aniqlansin.
- 2.029 Haqiqiy turga tegishli x o'zgaruvchi berilgan. Uning qiymatiga mos keluvchi f funksiyaning qiymati topilsin.

$$f(x) = \begin{cases} -x & x \leq 0 \\ x^2 & 0 < x < 2 \\ 4 & x \geq 2 \end{cases}$$

- 2.030** Haqiqiy turga tegishli x o'zgaruvchi berilgan. Uning qiymatiga mos keluvchi f funksiyaning qiymati topilsin.

$$f(x) = \begin{cases} 0, & x < 0 \quad x \in (-\infty; 0) \cup [4; +\infty), \\ 1, & x \in [0, 1) \cup [2, 3), \\ -1, & x \in [1, 2) \cup [3, 4). \end{cases}$$

- 2.031** Butun son berilgan. Uni satr ko'rinishida "manfiy juft son", "nol soni", "musbat toq son" va hokazo shaklda chiqarilsin.

- 2.032** Berilgan x uchun quyidagi ifodalar qiymatlari o'sish tartibida chop etilsin: chx , $1+|x|$ va $(1+x^2)^x$;

- 2.033** $a_1x+b_1y=c_1$ va $a_2x+b_2y=c_2$ tenglamalar bilan berilgan chiziqlarning kesishish nuqtasi koordinatalarini chop etadigan yoki bu chiziqlarning kesishmasligi yoki ustma-ust tushishligi yoki parallelligi haqida ma'lumot beradigan dastur tuzilsin. Bu yerda $a_1, b_1, c_1, a_2, b_2, c_2$ -berilgan sonlar.

- 2.034** $ax^4+bx^2+c=0$ tenglamaning haqiqiy ildizlarini topadigan yoki ildizi yo'qligi haqida ma'lumot beradigan dastur tuzilsin.

- 2.035** Agar tomonlarining uzunliklari ixtiyoriy a, b, s sonlarga teng bo'lgan uchburchakni qurish mumkin bo'lmasa 0, aks holda uchburchak teng tomonli bo'lsa 3, teng yonli bo'lsa 2 va boshqa hollar uchun 1 qiymatini chop qiluvchi dastur tuzilsin.

- 2.036** Ikki xonali sonlar ketma-ketligi 1011121314...9899 berilgan bo'lib, uning k - o'rindagi ($1 \leq k \leq 180$) raqami aniqlansin.

- 2.037** 10 sonining darajalaridan tuzilgan ketma-ketlik 101001000... berilgan bo'lib, uning k -o'rindagi raqami aniqlansin. Bu yerda k natural son.

- 2.038** Berilgan 50 ta haqiqiy sonlarning eng kattasini topadigan dastur tuzilsin.

- 2.039** Berilgan 50 ta haqiqiy sonlarning eng kichigini topadigan dastur tuzilsin.

- 2.040** Butun n soni ($n>0$) va n ta haqiqiy sonlar berilgan. Ular orasidan manfiylari soni topilsin.

- 2.041** Bo'sh bo'lmagan va oxiri 0 soni bilan tugaydigan musbat butun sonlar ketma-ketligi berilgan (0 ketma-ketlikka kirmaydi

va uning tugaganligini bildiradi). Ketma-ketlikning o'rta geometrik qiymatini hisoblaydigan dastur tuzilsin.

2.042 a, b sonlarining eng katta umumiy bo'luvchisi - s topilsin.

2.043 $\cos(ctgn)$, $n=1,2,3...$ sonlar ketma-ketligining birinchi manfiy hadi topilsin.

2.044 Hisoblansin:

$$p = \left(1 - \frac{1}{2^2}\right) \left(1 - \frac{1}{3^2}\right) \dots \left(1 - \frac{1}{n^2}\right), \quad n > 2;$$

2.045 Hisoblansin:

$$p = \cos(1 + \cos(2 + \dots + \cos(39 + \cos 40) \dots)).$$

2.2. Tanlash operatori

Shart operatorining yana bir ko'rinishi *switch* tarmoqlanish (*tanlash*) operatori bo'lib, uning sintaksisi quyidagicha:

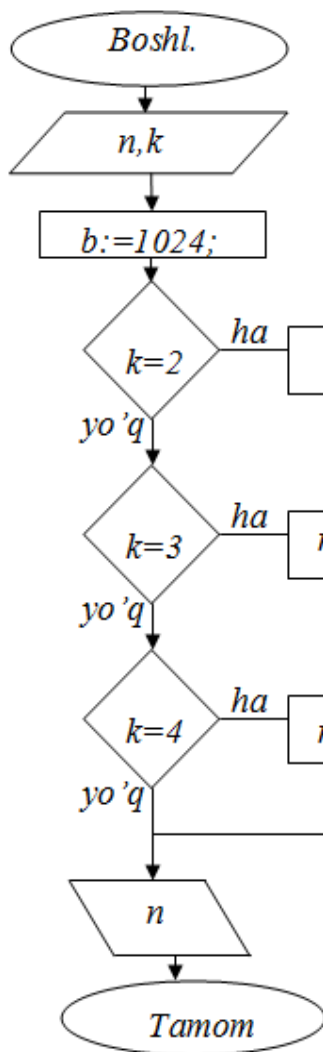
```
switch (<ifoda>)
{
case <o'zgarmas ifoda 1>: <operatorlar guruhi 1>; break;
case <o'zgarmas ifoda 2> : <operatorlar guruhi 2>; break;
...
case <o'zgarmas ifoda n> : <operatorlar guruhi n>; break;
default: <operatorlar guruhi n+1>;
}
```

Bu operator quyidagi tartibda ishlaydi: birinchi navbatda *<ifoda>* qiymati hisoblanadi, keyin bu qiymat *case* kalit so'zi bilan ajratilgan *<o'zgarmas ifoda i>*lar bilan ($i=1..n$) taqqoslanadi. Agar ifoda qiymati *i*-ifoda qiymatiga teng bo'lsa, shu qatordagi ':' belgisidan boshlab, toki *break* kalit so'zigacha bo'lgan *<operatorlar guruhi>* bajariladi va boshqaruv tarmoqlanuvchi (*tanlash*) operatoridan keyingi joylashgan operatorga o'tadi. Agar *<ifoda>* birorta ham *<o'zgarmas ifoda>*lar bilan mos kelmasa, qurilmaning *default* qismidagi *<operatorlar guruhi n+1>* bajariladi. Shuni qayd etish kerakki, qurilmada *default* kalit so'zi faqat bir marta ishlatiladi.

Umuman olganda, tarmoqlanuvchi (*tanlash*) operatorida *break*, *default* kalit so'zlari ishlatilmasligi ham mumkin. Lekin bu holda operatorning mazmuni o'zgarishi mumkin. Masalan, *default* qismi bo'lmagan holda, agar *<ifoda>* birorta *<o'zgarmas ifoda>*lar bilan ustma-ust tushmasa, operator hech qanday amal bajarmasdan

boshqaruvni tarmoqlanuvchi (*tanlash*) operatoridan keyingi operatorga uzatadi. Agar *break* kalit soʻzi ishlatilmasa, *<ifoda>* birorta *<oʻzgarmas ifoda i>* bilan ustma-ust tushgan holda, unga mos keluvchi operatorlar guruhini bajaradi va «*toʻxtamasdan*» keyingi qatordagi operatorlar guruhini bajarishga oʻtib ketadi.

Masala. Axborot oʻlchov birliklari quyidagi koʻrinishda nomerlangan: 1-bayt, 2-kilobayt, 3-megabayt, 4-gigabayt. Axborot oʻlchovi n va uning birligi k butun ($k \in [1-4]$) sonlari berilgan. Berilgan axborotning baytdagi qiymatini hisoblaydigan algoritim va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char
*argv[])
{
    int n, k;
    /*Axborot hajmi va
    birligini kiritish*/
    cin >> n >> k;
    int b = 1024;
    /*k ning qiymatiga mos
    amalni bajarish*/
    switch (k)
    {
        case 2: n *= b; break;
        case 3: n *= b * b; break;
        case 4: n *= b * b * b;
    }
    //Natijani chiqarish
    cout << "n=" << n;
    system("PAUSE");
    return EXIT_SUCCESS;
}
  
```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 2.046 1-7 oraligʻidagi butun son berilgan boʻlsa, shu songa mos keladigan hafta kunining nomi ekranga chiqarilsin.
- 2.047 Bahoni anglatadigan k soni berilgan. Agar $k=1$ boʻlsa “yomon”,

$k=2$ bo'lsa "qoniqarsiz", $k=3$ bo'lsa "qoniqarli", $k=4$ bo'lsa "yaxshi", $k=5$ bo'lsa "a'lo" so'zlaridan birini, agar k bu sonlardan hech biriga teng bo'lmasa "xato" so'zi chop etilsin.

- 2.048** Oy nomerini ifodalaydigan 1-12 oraliqdagi son berilgan. Shu nomerga tegishli oyning qaysi faslga tegishli ekanligi aniqlansin.
- 2.049** Oy nomerini ifodalaydigan 1-12 oraliqdagi son berilgan. Shu nomerdagi oyga mos keluvchi (kabisa bo'lmagan yildagi) kunlar soni aniqlansin.
- 2.050** Arifmetik amallar quyidagi ko'rinishda nomerlangan. 1 "qo'shish", 2 "ayirish", 3 "ko'paytirish", 4 "bo'lish". a va b ($b \neq 0$) haqiqiy sonlari hamda n butun soni ($n \in [1, 4]$) berilganda n ga mos keluvchi amal bajarilsin va natija chop etilsin.
- 2.051** Uzunlik birliklar quyidagi ko'rinishda nomerlangan: 1-km, 2-m, 3-dm, 4-sm, 5-mm. Kesma uzunligi l (haqiqiy) va uning birligi n butun ($n \in [1; 5]$) sonlari berilgan bo'lsa, uning metrdaqi qiymati hisoblansin.
- 2.052** Massa birliklari quyidagi ko'rinishda nomerlangan: 1-mg, 2-gr, 3-kg, 4-sentr, 5-tonna. Jism massasi m (haqiqiy) va uning birligi n butun ($n \in [1; 5]$) sonlari berilgan bo'lsa, uning kg dagi qiymati hisoblansin.

B toifa

- 2.053** To'g'ri sanani ifodalovchi kabisa bo'lmagan yilning (d) kuni va (m) oyi butun sonlar ko'rinishida berilgan. Berilgan sanadan oldingi sana aniqlansin.
- 2.054** To'g'ri sanani ifodalovchi kabisa bo'lmagan yilning (d) kuni va (m) oyi butun sonlar ko'rinishida berilgan. Berilgan sanadan keyingi sana aniqlansin.
- 2.055** Robot to'rtta yo'nalish bo'yicha harakatlanishi mumkin. ("1" Shimol, "2" G'arb, "3" Janub, "4" Sharq); va uch xil buyruqni qabul qila oladi, ya'ni 0 harakatni davom ettirish, 1-chapga burilish, -1 o'ngga burilish. Robotning yo'nalishi c va unga uzatilgan buyruq n berilgan bo'lsa, buyruqdan keyingi robotning holati aniqlansin.
- 2.056** Lokatr dunyoning biror tomoniga qaratilishga mo'ljallangan ("1" Shimol, "2" G'arb, "3" Janub, "4" Sharq) va u uch xil sonli burilish buyruqlarini qabul qiladi: 1 chapga burilish, -1 o'ngga burilish, 2 180° ga burilish. Dastlabki yo'nalish - c va n_1 hamda n_2 buyruqlar berilganda, lokatrning buyruqlar

- bajarilgandan keyingi yo'nalishi aniqlansin.
- 2.057** Doira elementlari quyidagi ko'rinishda nomerlangan: 1-radius, 2-diametr $d=2r$, 3-usunlik $l=2pr$, 4-yuza $s=pr^2$. Bu elementlardan birining nomeri va son qiymati berilgan bo'lsa, doiraning qolgan elementlari hisoblansin. ($p=3,14$)
- 2.058** Teng yonli to'g'ri burchakli uchburchak elementlari berilgan. 1-katet a , 2-gipotenuza c , $c=a\cdot\sqrt{2}$, 3-balandlik h , (gipotenuzaga tushirilgan) $h=\frac{c}{2}$, 4-yuza $s=\frac{c\cdot h}{2}$. Bu elementlarning birortasining nomeri va son qiymati berilganda, qolgan elementlarining qiymatlari hisoblansin.
- 2.059** Teng tomonli uchburchakning elementlari quyidagicha nomerlangan. 1-tomoni a , 2-radius r_1 , (uchburchakka ichki chizilgan aylananing radiusi $R_1=\left(\frac{\sqrt{3}}{6}\right)\cdot a$), 3- r_2 (uchburchakka tashqi chizilgan aylananing radiusi $r_2=2r_1$), 4-yuza $s=a^2\cdot\frac{\sqrt{3}}{4}$. Bu elementlarning birortasining nomeri va son qiymati berilganda, qolgan elementlarning qiymatlari hisoblansin.
- 2.060** Kartaning $6\leq k\leq 10$ dagi qismi o'z holicha va 10 dan yuqori bo'lgan qismi quyidagicha: 11-valit, 12-dama, 13-karol, 14-tuz nomerlangan. Gullari esa quyidagicha nomerlangan: 1-toppon, 2-g'isht, 3-chillik, 4-qarg'a. Ikkita butun k (kartaning tartibi, $6\leq k\leq 14$) va m (karta guli $1\leq m\leq 4$) sonlari berilganda, ularga mos keluvchi karta nomi aniqlansin.
- 2.061** Yoshni anglatuvchi 10-99 oraliqda bo'lgan butun son berilgan. Berilgan yosh so'z bilan ifodalansin.
- 2.062** 1-40 oralig'idagi o'quv masalasining tartib nomerini bildiruvchi butun son berilgan. Bu son satr ko'rinishda ifodalansin.
- 2.063** 100-999 oralig'idagi butun son berilgan. Bu son satr ko'rinishida ifodalansin.
- 2.064** Sharq kalendarida yil nomlari uchun 60 yillik sikl qabul qilingan bo'lib, u o'n ikki yillik 5 ta sikldan tashkil topgan va ular quyidagi ranglar bilan aniqlanadi: yashil, qizil, sariq, oq va qora. Har bir ichki sikldagi yil, hayvonlar nomi bilan nomlangan (sichqon, sigir, yo'lbars, quyon, ajdarho, ilon, ot, qo'y, maymun, tovuq, it va cho'chqa). Yil nomeri berilganda yuqoridagilardan foydalanib uning nomi aniqlansin.

- 2.065** To'g'ri sanani ifodalovchi ikkita d va m sonlari berilgan. Ularga mos keluvchi burj nomi aniqlansin: (*suvchi, baliq, qo'y, buzoq, egizaklar, qisqichbaqa, arslon, parizod, tarozi, chayon, yoy, tog' echkisi*). Burjlar quyidagicha aniqlanadi: 21-martdan 21-aprelgacha *suvchi* va hokazo...

2.3. Bobga doir qiyinroq masalalar

- 2.066** *Lift [11]*

M qavatli uyning N -qavatiga yangi shkafni olib chiqish kerak. Ali yuk tashuvchi brigadani yordamga chaqirdi. Yuk tashuvchilarga to'lov quyidagicha amalga oshiriladi: shkafni bitta qavat yuqoriga chiqarish uchun 2000 so'm, bitta qavat pastga tushirish uchun esa 1000 so'm haq to'lanadi. Shkafni liftga chiqarish va tushirishga haq to'lanmaydi. Shunga qaramasdan baribir Ali yuk tashuvchilarga eng kam haq to'lashi kerak. Lift birinchi qavatdan boshlab faqatgina har bir K -chi qavatda to'xtaydi (*ya'ni 1, $K+1$, $2K+1$, $3K+1$, ...*). Aliga birinchi qavatdan N -chi qavatga yuk tashuvchilarga kam haq to'lagan holda shkafni olib chiqishga yordam bering.

Berilganlar:

Bo'sh joylar bilan ajratilgan uchta son beriladi: M ($2 \leq M \leq 100$), N ($2 \leq N \leq M$) va K ($2 \leq K \leq M-1$).

Natija:

Shkafni olib chiqish uchun sarflanadigan eng kam ish haqi aniqlansin.

Berilganlar:	Natija:
20 7 4	200
20 7 2	0

- 2.067** *Yo'lovchilarni taqsimlash[11].*

Judayam tig'iz vaqtda bekatga bitta yo'nalishda qatnovchi uchta taksi keldi va ularga yo'lovchilar yopirilishdi. Haydovchilar har bir taksidagi yo'lovchilar soni har xil ekanligini sezib qolishdi. Yo'lovchilarni bir qismini boshqa taksiga olgan holda har bir taksida yo'lovchilar soni teng bo'linishini talab qilishdi. Shuning uchun eng kam sonli yo'lovchilarni almashtirish orqali taksilardagi yo'lovchilar sonini tenglashtirish zarur.

Berilganlar:

Har birida 100 dan oshmaydigan mos ravishda birinchi, ikkinchi

va uchinchi taksilardagi yo'lovchilar soni berilgan
Natija:

Taksilaradagi yo'lovchilar miqdorini tenglashtirish uchun bajariladigan eng kam almashtirishlar soni chiqarilsin. Agarda bunday qilishni iloji bo'lmasa IMPOSSIBLE (katta harflar bilan) so'zini chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
1 2 3	1
99 100 100	IMPOSSIBLE

2.068

Mehmonxona[11].

Fan olimpiadalarining final bosqichiga respublikamizning har bir viloyatidan mehmonlar kelishadi. Ularni mehmonxonaga tartib bilan joylashtirish oson masala emas. Kelgan ishtirokchilar soni n ta. Mehmonxonada ularni joylashtirish uchun faqatgina ikki va uch kishilik xonalar mavjud. Mehmonlar pulni tejash maqsadida kam sonli xonalarni band qilishga kelishib oldilar, ya'ni band qilingan xonada bo'sh o'rin bo'lmasligi lozim. n kishidan iborat bo'lgan mehmonlarni ikki va uch kishilik xonalarga shunday joylashtiringki, xonalar soni eng minimal bo'lsin. Buni aniqlashda ularga yordam bering.

Berilganlar:

Mehmonlar sonini bildiruvchi n ($2 < n < 100$) soni berilgan.

Natija:

Bo'sh joy bilan ajratilgan mehmonlar joylashtirilgan ikki va uch kishilik xonalar soni chiqarilsin. Dastlab ikki kishilik, so'ngra uch kishilik.

<i>Berilganlar:</i>	<i>Natija:</i>
7	2 1
2	1 0

2.069

Pitsa[11].

Bitta guruhda o'qiydigan M ta talaba pitsaxo'rlikka kelishdi. Ular maslahatlashib, K bo'lakli bitta katta pitsaga buyurtma berishdi. Pitsa doira shaklida bo'lib, teng K ta sektorlarga bo'lingan. Ularning har biri o'z navbatida qiymalardan iborat. Pitsa kesilmagan holda beriladi. Ular ofisiantdan pitsani har birlariga bittadan M ta teng sektorlarga bo'lib berishni so'rashdi. Bunda ofisiant shunday bo'lishi kerakki, har bitta odamga hech bo'lmaganda bo'lakda ikkita qiyma bo'lsin. Ofisiant agarda pitsani eng optimal ko'rinishda kessa nechta

odamga bittadan ko'p qiyma tegishini aniqlashga yordam bering.

Berilganlar:

K, M ikkita son berilgan ($1 \leq K \leq 100, 1 \leq M \leq 100$). Mos ravishda pitsadagi bo'laklar va talabalar soni.

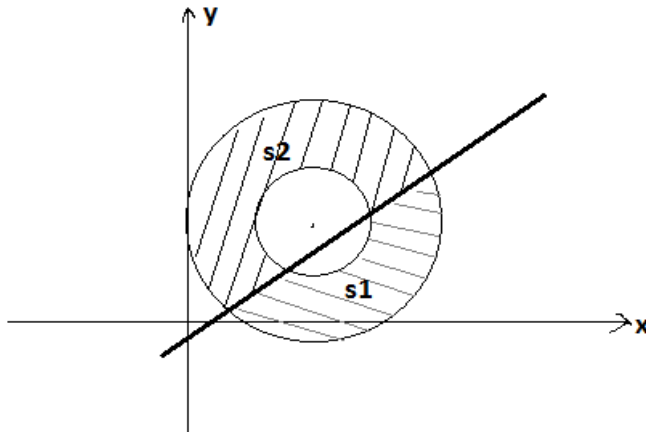
Natija:

Bittadan ortiq qiymaga ega bo'ladigan eng ko'p talabalar soni chiqarilsin.

2.070

Xalqa bo'lagi.

Ikki o'lchamli koordinatalar sistemasida markazi (x_0, y_0) nuqatada, ichki radiusi r_1 va tashqi radiusi r_2 ga teng bo'lgan xalqa berilgan. Tekislik orqali (x_1, y_1) va (x_2, y_2) nuqtalar orqali o'tuvchi to'g'ri chiziq o'tkazilgan. To'g'ri chiziq xalqani kesib o'tadi. Xalqa ikkita qismga ajralgan. Bu ikki qismning yuzalarini toping.



Berilganlar:

Birinchi qatorda 8 ta butun son – $x_0, y_0, r_1, r_2, x_1, y_1, x_2, y_2$ sonlari berilgan ($1 \leq r_1 < r_2 \leq 5000, -5000 \leq x_0, y_0, x_1, y_1, x_2, y_2 \leq 5000$). (x_1, y_1) va (x_2, y_2) nuqtalar ustma-ust tushmaydi. To'g'ri chiziq xalqa bilan kamida bitta umumiy nuqtaga ega.

Natija:

Xalqaning to'g'ri chiziq orqali ajralgan ikkita qismning yuzalarini 10^{-3} aniqlikda bitta bo'sh joy bilan ajratib chiqaring. Avval kichik qismning yuzasi chiqarilsin (qismlar teng bo'lsa, buning ahamiyati yo'q).

<i>Berilganlar:</i>	<i>Natija:</i>
5 4 2 5 -1 -1 11 8	30.580 35.393

2.071

Kotlet [10]

”Dasturlash asoslari” fanidan yakuniy nazorat topshirgan talaba xursand. Shuning uchun u, misol ishlashda ”yordam” bergan

guruhdoshlariga kotlet olib berishga qaror qildi. Ular oshxonaga borib N ta kotletga buyurtma berishdi. Kotlet tayyorlash uchun oshpaz uning har bir tomonini tovada bir minut davomida pishiradi. Afsuski, oshpazni bitta tovasi bor. Unda bir vaqtning o'zida eng ko'pi bilan K ta kotletni pishirish mumkin. Talabalar buyurtmasini bajarishi uchun oshpaz sarflaydigan eng kam vaqtni aniqlashga yordam bering.

Berilganlar:

Bitta satrda N va K ($1 \leq N, K \leq 1000$) sonlari berilgan. N - talabalar soni, K esa tovaga sig'adigan kotletlar soni.

Natija:

N ta kotletni tayyorlash uchun oshpaz sarflaydigan eng kam vaqt (minutlarda) chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
3 2	3
5 3	4
12 1	24

III Bob. Takrorlanuvchi algoritmlar va ularni dasturlash

Dastur bajarilishini boshqarishning yana bir kuchli mexanizmlaridan biri – takrorlash operatorlari hisoblanadi.

Takrorlash operatori «*takrorlash sharti*» deb nomlanuvchi ifodaning rost qiymatida dasturning ma'lum bir qismidagi operatorlar (takrorlash tanasi)ning ko'p marta takror ravishda bajarilishini amalga oshiradi.

Takrorlash o'zining kirish va chiqish nuqtalariga ega bo'lib, ba'zan chiqish nuqtasi bo'lmasligi ham mumkin. Agar chiqish nuqtasi bo'lmasa cheksiz takrorlanish sodir bo'ladi. Cheksiz takrorlanish uchun takrorlashni davom ettirish sharti doimo rost bo'ladi.

Takrorlash shartini tekshirish, takrorlanish tanasidagi operatorlarning bajarilishidan oldin tekshirilishi mumkin (*for*, *while* takrorlash operatorlarida) yoki takrorlanish tanasining operatorlari bir marta bajarilgandan keyin tekshirilishi mumkin (*do-while*).

Takrorlanishlarni tashkil etishda takrorlanishlar soni va qadami oldindan ma'lum bo'lgan holatlar uchun *for* operatoridan, boshqa hollarda esa *while* hamda *do-while* operatorlaridan foydalanish maqsadga muvofiq.

Takrorlash operatorlari ichma-ich joylashishi ham mumkin. Quyida takrorlash operatorlari va ularning qo'llanilishi bilan batafsil tanishamiz.

3.1. *for* operatoridan foydalanib takrorlash jarayonlarini tashkil etish

for takrorlash operatorining sintaksisi quyidagi ko'rinishga ega:

for (<*ifoda1*>; <*ifoda2*>; <*ifoda3*>) <*operator yoki blok*>;

Bu operator o'z ishini <*ifoda1*> ifodasining bajarilishidan boshlaydi. Keyin takrorlash qadamlari boshlanadi. Har bir qadamda <*ifoda2*> tekshiriladi, agar <*ifoda2*> ning qiymati 0 qiymatidan farqli, ya'ni rost bo'lsa, takrorlash tanasi – <*operator yoki blok*> bajariladi, so'ngra <*ifoda3*> bajariladi. Agar <*ifoda2*> ning qiymati 0 (*yo'lg'on*) bo'lsa, takrorlanish jarayoni to'xtaydi va boshqaruv takrorlash operatoridan keyingi operatorga o'tadi. Shuni ta'kidlash kerakki, <*ifoda2*> ifodasi vergul bilan ajratilgan bir nechta ifodalar birlashmasidan iborat bo'lishi ham mumkin, bu holda vergul bilan ajratilgan ifodalarning oxirigisi takrorlanish sharti hisoblanadi. Takrorlanish tanasi sifatida bitta operator,

jumladan bo'sh operator bo'lishi yoki operatorlar bloki ham kelishi mumkin.

Quyida *for* operatori parametrlarining vazifalari bilan tanishamiz:

<ifoda1> – takrorlash sanagichiga boshlang'ich qiymat berishga xizmat qiladi va u takrorlash jarayoni boshida faqat bir marta hisoblanadi. Ifodada o'zgaruvchi e'loni ham bo'lishi mumkin va bu o'zgaruvchi takrorlash operatori tanasida amal qiladi va takrorlash operatoridan tashqarida «ko'rinmaydi».

<ifoda2> – takrorlanishning bajarishi yoki bajarilmasligini aniqlab beruvchi mantiqiy ifoda (*shart*), agar shart *rost* bo'lsa, takrorlanish davom etadi, aks holda to'xtaydi. Agar bu ifodaning o'zni bo'sh bo'lsa, shart doimo *rost* deb hisoblanadi.

<ifoda3> – odatda takrorlanish sanagichining qiymatini oshirish (kamaytirish) uchun xizmat qiladi yoki unda takrorlanish shartiga ta'sir etuvchi boshqa amallar ham bo'lishi mumkin.

Takrorlash operatorida qavs ichidagi ifodalar bo'lmasligi ham mumkin, lekin sintaksis ';' bo'lmasligiga ruxsat bermaydi. Shu sababli sodda ko'rinishdagi takrorlash operatori quyidagicha bo'ladi:

for(;;).

Agar takrorlanish jarayonida bir nechta o'zgaruvchilarning qiymati sinxron ravishda o'zgarishi kerak bo'lsa, <ifoda1> va <ifoda3> ifodalarida zarur operatorlarni ';' bilan yozish orqali bunga erishish mumkin:

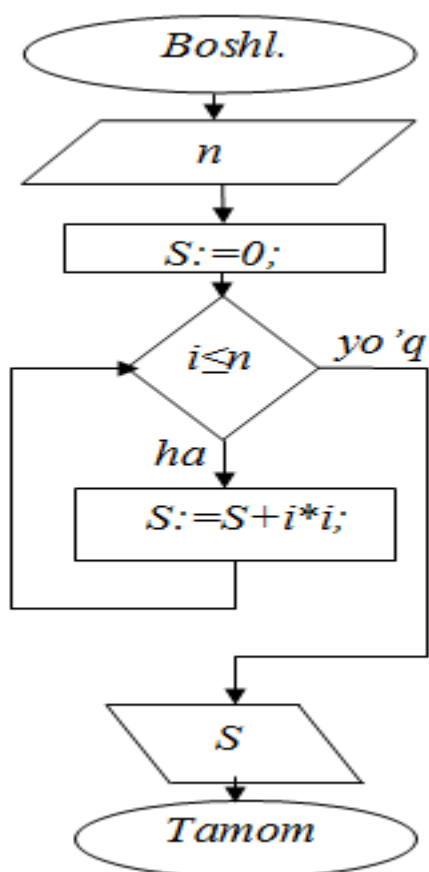
for(int *i*=1,*j*=2;*i*<=15;*i*++,*j*=*i*+1) { ... }

Takrorlash operatorining har bir qadamida *i* va *j* o'zgaruvchilarning qiymatlari mos ravishda o'zgarib boradi.

for operatorida takrorlanish tanasi bo'lmasligi ham mumkin.

Shuni ham ta'kidlash joizki, C++ tilida *for* operatorining parametrlari haqiqiy turga tegishli bo'lishi, takrorlanishlar qadami ham ixtiyoriy holatda belgilanishi mumkin. Bu imkoniyatlar operatorning keng qamrovli ekanligini bildiradi.

Masala. Dastlabki n ta ($n > 0$) natural sonlarning kvadratlari yig'indisini hisoblovchi algoritm va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char
*argv[])
{
    int i,n;
    //Berilgan qiymatni kiritish
    cin>>n;
    /*Yig'uvchiga boshlang'ich
    qiymat berish*/
    int s=0;
    /*1 dan n gacha bo'lgan sonlar
    kvadratlari yig'ish*/
    for (i=1;i<=n;i++)
        s+=i*i;
    //Natijani chiqarish
    cout<<"S="<<s;
    system("PAUSE");
    return EXIT_SUCCESS;
}
  
```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 3.001 k va n ($n > 0$) butun sonlar berilgan. n marta k soni chiqarilsin.
- 3.002 a va b butun sonlar berilgan ($a < b$). Ular orasida joylashgan butun sonlar (a va b sonlari ham kiradi) o'sish tartibida chiqarilsin hamda shu sonlar miqdori (soni) hisoblansin.
- 3.003 a va b butun sonlar berilgan ($a < b$). Ular orasida joylashgan butun sonlar (a va b sonlari ham kiradi) kamayish tartibida chiqarilsin hamda shu sonlar miqdori (soni) hisoblansin.
- 3.004 1 kg konfetning narxi haqiqiy sonda berilgan. 1,2,..., 10 kg konfetlarning bahosi chiqarilsin.
- 3.005 1 kg konfetning narxi berilgan. 0.1, 0.2, ..., 1 kg konfetlarning bahosi chiqarilsin.
- 3.006 1 kg konfetning narxi berilgan. 1.2, 1.4, ..., 2 kg konfetlarning bahosi chiqarilsin.
- 3.007 a va b butun sonlar berilgan ($a < b$). a dan b gacha bo'lgan (a va b sonlari ham kiradi) butun sonlar yig'indisi hisoblansin.
- 3.008 a va b butun sonlar berilgan ($a < b$). a dan b gacha bo'lgan (a va

- 3.009** a va b butun sonlar berilgan. ($a < b$) a dan b gacha bo'lgan (a va b sonlari ham kiradi) sonlarning kvadratlari yig'indisi hisoblansin.
- 3.010** n butun ($n > 0$) soni berilgan. $1 + \frac{1}{2} + \dots + \frac{1}{n}$ yig'indi hisoblansin (Natija haqiqiy son ko'rinishida bo'ladi).
- 3.011** n butun ($n > 0$) soni berilgan $n^3 + (n+1)^3 + (n+2)^3 \dots + (2n)^3$ yig'indi hisoblansin (Natija haqiqiy son ko'rinishida bo'ladi).
- 3.012** n ($0 < n < 10$) butun soni berilgan. $1, 1-1, 2+1, 3-\dots$ Ifodaning qiymati hisoblansin. Shart operatori qo'llanilmasin.
- 3.013** n butun ($n > 0$) soni berilgan. Quyidagi formuladan foydalanib berilgan sonning kvadrati hisoblansin: $n^2 = 1 + 3 + 5 + \dots + (2n-1)$. Har bir qadamdagi yig'indi chiqarilsin (natijada 1 dan n gacha bo'lgan butun sonlarning kvadrati chiqadi).
- 3.014** a haqiqiy va n butun sonlari berilgan ($n > 0$). $a^n = a \cdot a \cdot \dots \cdot a$ a ning n - darajasi hisoblansin.
- 3.015** a haqiqiy va n butun ($n > 0$) sonlari berilgan. Bitta sikldan foydalanib a sonining 1 dan n gacha bo'lgan darajalari hisoblab chiqarilsin.
- 3.016** a haqiqiy va n butun ($n > 0$) sonlari berilgan. $1 + a + a^2 + a^3 + \dots + a^n$. Bitta sikldan foydalanib yig'indi hisoblansin.
- 3.017** a haqiqiy va n butun ($n > 0$) sonlari berilgan. $1 - a + a^2 - a^3 + \dots + (-1)^n a^n$. Bitta sikldan foydalanib ifodaning qiymati hisoblansin. Hisoblashda shart operatoridan foydalanilmasin.
- 3.018** n butun ($n > 0$) son berilgan. $n! = 1 \cdot 2 \cdot \dots \cdot n$ (n -faktorial) ko'paytma hisoblansin. Ifodaning natijasi butun sonlar diapazonidan chiqib ketishi mumkinligi hisobga olinib, natijani saqlash uchun haqiqiy turga tegishli o'zgaruvchidan foydalanilsin va natija ham haqiqiy son ko'rinishida chiqarilsin.
- 3.019** n butun ($n > 0$) soni berilgan. Bitta sikldan foydalanib $1! + 2! + \dots + n!$ yig'indi hisoblansin.
- 3.020** Sonlar o'qida a, b haqiqiy ($a < b$) sonlari va n butun soni ($n > 1$) berilgan. $[a, b]$ kesma n ta teng kesmachalarga bo'linganda, ularning uzunligi h hamda $(a, a+h, a+2h, \dots, b)$. $[a, b]$ kesmani bo'lishdan hosil bo'lgan nuqtalar chiqarilsin.
- 3.021** n butun son va sonlar o'qida a, b ($a < b$) haqiqiy sonlar berilgan. $[a, b]$ kesma n ta teng kesmachalarga ajratilgan. Har

bir kesma uzunligi h hamda $[a,b]$ kesmani bo'luvchi nuqtalardagi $f(x)=1-\sin(x)$ funksiyaning qiymati chiqarilsin.

- 3.022** n butun ($n>0$) son berilgan. Haqiqiy turga tegishli a_k ketma-ketlik quyidagicha aniqlanadi.

$$a_0=2; \quad a_k = 2 + \frac{1}{a_{k-1}} \quad k=1,2,\dots \quad \text{Ketma-ketlikning } a_1, a_2, \dots, a_n$$

elementlari hisoblansin.

- 3.023** n butun ($n>0$) son berilgan. Haqiqiy turga tegishli a_k ketma-ketlik quyidagicha aniqlanadi.

$$a_0=1; \quad a_k = \frac{a_{k-1}+1}{k}, \quad k=1,2,\dots \quad \text{Ketma-ketlikning } a_1, a_2, \dots, a_n$$

elementlari hisoblansin.

- 3.024** n butun ($n>1$) son berilgan. Butun turga tegishli f_k fibonachchi sonlar ketma-ketligi quyidagicha aniqlanadi. $f_1=1; f_2=1; f_k=f_{k-2}+f_{k-1}, k=3,4,\dots f_1, f_2, \dots, f_n$ elementlari hisoblansin.

- 3.025** n butun ($n>1$) son berilgan. Haqiqiy turga tegishli a_k ketma-ketlik quyidagicha aniqlanadi.

$$a_1=1; a_2=2; \quad a_k = \frac{a_{k-2} + 2 \cdot a_{k-1}}{2}, \quad k=3,4,\dots$$

$a_1, a_2, \dots, a_n$ elementlari hisoblansin.

- 3.026** n butun ($n>3$) son berilgan. a_k butun sonli ketma-ketlik quyidagicha aniqlanadi. $a_1=1; a_2=2; a_3=3 \quad a_k=a_{k-1}+a_{k-2}-2a_{k-3}, k=4,5,\dots$

$a_1, a_2, \dots, a_n$ elementlari hisoblansin.

- 3.027** n ta haqiqiy ($n>1$) sonlardan iborat ketma ketlik berilgan. Berilgan ketma ketlikdagi sonlarning butun qismlari yig'indisi hisoblansin.

- 3.028** n ta musbat haqiqiy ($n>1$) sonlar ketma ketligi berilgan. Berilgan ketma ketlikdagi sonlarning kasr qismlari ko'paytmasi hisoblansin.

- 3.029** n ta haqiqiy ($n>1$) sonidan iborat ketma ketlik berilgan. Berilgan ketma ketlikdagi sonlarning yaxlitlangan qiymatlari yig'indisi hisoblansin.

- 3.030** n ta butun ($n>1$) sonidan iborat ketma ketlik berilgan. Berilgan ketma ketlikdagi hamma juft sonlar va ularning miqdori k hisoblansin.

B toifa

- 3.031** n butun ($n > 0$) soni berilgan. $1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$. Bitta sikldan foydalanib, yig'indi hisoblansin.
- 3.032** x haqiqiy va n butun ($n \geq 0$) sonlari berilgan. $1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$. Ifodaning qiymati hisoblansin.
- 3.033** $x \in \mathbb{R}$ va $n \in \mathbb{Z}$ sonlari berilgan. $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots - \frac{(-1)^n x^{2n-1}}{(2n-1)!}$. Ifodaning qiymati hisoblansin.
- 3.034** x haqiqiy va n butun ($n \geq 0$) sonlari berilgan. $1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots - \frac{(-1)^n x^{2n}}{(2n)!}$. Ifodaning qiymati hisoblansin.
- 3.035** x haqiqiy va n butun ($n > 0$) sonlari berilgan. $x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^{n-1} x^n}{n}$. Ifodaning qiymati hisoblansin.
- 3.036** x haqiqiy ($|x| < 1$) va n butun ($n > 0$) sonlari berilgan. $x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + \frac{(-1)^{n-1} x^{2n-1}}{2n-1}$. Ifodaning qiymati hisoblansin.
- 3.037** x haqiqiy ($|x| < 1$) va n butun ($n > 0$) sonlari berilgan. $x + \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 x^5}{2 \cdot 4 \cdot 5} + \dots + \frac{1 \cdot 3 \cdot \dots \cdot (2n-1) x^{2n-1}}{2 \cdot 4 \cdot \dots \cdot 2n \cdot (2n+1)}$. Ifodaning qiymati hisoblansin.
- 3.038** x haqiqiy ($|x| < 1$) va n butun ($n > 0$) sonlari berilgan. $1 + \frac{x}{2} - \frac{x^3}{2 \cdot 4} + \frac{1 \cdot 3 x^3}{2 \cdot 4 \cdot 6} + \dots + (-1)^n \cdot 1 \cdot 3 \cdot \dots \cdot \frac{(2n-3) x^{n-1}}{2 \cdot 4 \cdot 2n}$. Ifodaning qiymati hisoblansin.
- 3.039** n va k musbat butun sonlari berilgan. $1^k + 2^k + \dots + n^k$ yig'indi hisoblansin. Yig'indini hisoblashda butun turga tegishli o'zgaruvchi to'lib qolishi ya'ni, diapazondan chiqib ketishi mumkin. Shuning uchun yig'indini hisoblashda haqiqiy turga tegishli o'zgaruvchidan foydalanish maqsadga muvofiq.
- 3.040** n musbat butun son berilgan. $1^1 + 2^2 + \dots + n^n$ yig'indi hisoblansin. Yig'indini hisoblashda butun turga tegishli o'zgaruvchi to'lib qolishi ya'ni, diapazondan chiqib ketishi mumkin. Shuning uchun yig'indini hisoblashda haqiqiy turga tegishli o'zgaruvchidan foydalanish maqsadga muvofiq.

- 3.041** n musbat butun soni berilgan. $1^n + 2^{n-1} + \dots + n^1$ yig'indini hisoblang. Yig'indini hisoblashda butun sonli tur to'lib qolishi mumkin, ya'ni diapazondan chiqib ketishi mumkin. Shuning uchun haqiqiy o'zgaruvchi e'lon qilib natija chiqarilsin
- 3.042** a va b ($a < b$) musbat butun sonlar berilgan. a dan b gacha bo'lgan barcha butun (a va b ham kiradi) sonlar son qiymati nechaga teng bo'lsa, shuncha martadan chiqarilsin. Masalan: 3 sonini 3 marta.
- 3.043** a va b ($a < b$) butun sonlar berilgan. a dan b gacha bo'lgan barcha butun (a va b ham kiradi) sonlar quyidagicha chiqarilsin: a ni 1 marta, $a+1$ ni 2 marta va hokazo.
- 3.044** n ta ($n > 0$) butun sondan iborat ketma ketlik berilgan. Agar ketma ketlik musbat sonlardan iborat bo'lsa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.045** k musbat butun son va n ta ($n > 0$) butun sonlardan iborat ketma ketlik berilgan. Agar ketma ketlik k dan kichik sonlardan iborat bo'lsa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.046** n ta ($n > 0$) sondan iborat ketma ketlik berilgan. Berilgan ketma ketlikdagi o'zining chap yonida turgan elementdan kichik bo'lgan elementlar va ularning miqdori k chiqarilsin.
- 3.047** n ta sondan iborat ketma ketlik berilgan. Ketma ketlikdagi o'zining o'ng yonida joylashgan elementdan kichik bo'lgan elementlar va ularning miqdori k chiqarilsin.
- 3.048** n ta ($n > 0$) haqiqiy sondan iborat ketma ketlik berilgan. Agar berilgan ketma ketlik o'suvchi bo'lsa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.049** n ta ($n > 0$) haqiqiy sondan iborat ketma ketlik berilgan. Agar berilgan ketma ketlik kamayuvchi bo'lsa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.050** n ta ($n > 0$) haqiqiy sondan iborat ketma ketlik berilgan. Ketma ketlikning ichki elementlaridan hech bo'lmaganda bittasi chetki elementlaridan katta bo'lsa 0 (yolg'on), aks holda 1 (rost) chiqarilsin.
- 3.051** Kamida 2 ta nolga ega bo'lgan n ta ($n > 1$) butun sondan iborat ketma ketlik berilgan. Berilgan ketma ketlikdagi oxirgi 2 ta uchragan nollar orasida joylashgan sonlar yig'indisi chiqarilsin. Agar oxirgi nollar qatorasiga kelsa 0 (nol) chiqarilsin.
- 3.052** Kamida 2 ta nolga ega bo'lgan n ta ($n > 1$) butun sondan iborat ketma ketlik berilgan. Birinchi va oxirgi nollar orasida

- joylashgan sonlar yig'indisi chiqarilsin. Agar birinchi va oxirgi nol qatorasiga kelsa 0 (nol) chiqarilsin.
- 3.053** k butun soni va $(a_1, a_2, \dots, a_n)$ n ta ($n > 0$) haqiqiy sondan iborat ketma ketlik berilgan. Berilgan ketma ketlikdagi sonlarning har biri uchun k -daraja hisoblansin.
 $(a_1)^k, (a_2)^k, \dots, (a_n)^k$
- 3.054** $a_1, a_2, \dots, a_n$ haqiqiy sonlar ketma ketligi berilgan. Sonlar quyidagicha chiqarilsin:
 $a_1, a_2^2, \dots, a_{n-1}^{n-1}, a_n^n$
- 3.055** $a_1, a_2, \dots, a_n$ lardan iborat haqiqiy sonlar ketma ketligi berilgan. Sonlar quyidagicha chiqarilsin: $a_1^n, a_2^{n-1}, \dots, a_{n-1}^2, a_n$
- 3.056** Har birida n tadan ($n > 0$) element bo'lgan k ta ($k > 0$) butun sonli ketma ketliklar berilgan. Berilgan ketma ketliklarning har biri uchun elementlar yig'indisi chiqarilsin.
- 3.057** Har birida n tadan ($n > 0$) element bo'lgan k ta ($k > 0$) butun sonli ketma ketliklar berilgan. Har bir ketma ketlik uchun elementlar ko'paytmasi chiqarilsin.
- 3.058** Har birida n tadan ($n > 0$) element bo'lgan k ta ($k > 0$) butun sonli ketma ketliklar berilgan. 2 soniga ega bo'lgan ketma ketliklar miqdori topilsin. Agar bunday ketma ketlik yo'q bo'lsa 0 (nol) chiqarilsin.
- 3.059** Har birida n tadan ($n > 0$) element bo'lgan k ta ($k > 0$) butun sonli ketma ketliklar berilgan. Berilgan har bir ketma ketlikda 2 yoki 0 elementi uchramasa, uning 1-elementi, aks holda 0 chiqarilsin.
- 3.060** Har birida n tadan ($n > 0$) element bo'lgan k ta ($k > 0$) butun sonli ketma ketliklar berilgan. Berilgan har bir ketma ketlikda 2 yoki 0 elementi bo'lsa, ularning oxirgi elementi, aks holda 0 chiqarilsin.

3.2. *while* va *do-while* operatorlaridan foydalanish

while takrorlash operatori, operator yoki blokni takrorlanish sharti yolg'on (0) bo'lguncha takrorlashni amalga oshiradi. Uning sintaksisi quyidagicha:

while (<ifoda>) <operator yoki blok>;

Agar <ifoda> qiymati doimo *rost* bo'lsa, takrorlanish jarayoni cheksiz davom etadi. Xuddi shunday, <ifoda> takrorlanish boshlanishida *rost* bo'lib, uning qiymatiga takrorlanish tanasidagi hisoblash ta'sir

etmasa, ya'ni uning qiymati o'zgarmasa, bu holda ham takrorlanish cheksiz davom etadi.

while takrorlanish shartini oldindan tekshiruvchi takrorlash operatori hisoblanadi. Agar takrorlanish boshida <ifoda> yolg'on bo'lsa, *while* operatori tarkibidagi <operator yoki blok> bajarilmasdan cheklab o'tiladi.

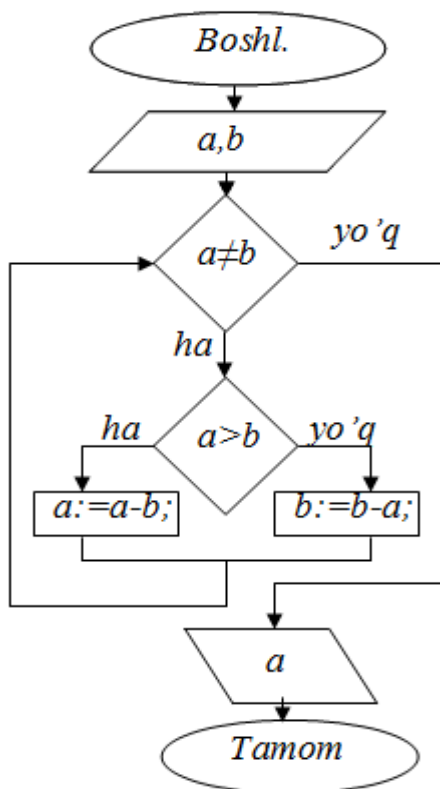
Ayrim hollarda <ifoda> qiymat berish operatori ko'rinishida kelishi mumkin. Bunda qiymat berish amali bajariladi va natija 0 bilan taqqoslanadi. Natija noldan farqli bo'lsa, takrorlanish davom ettiriladi.

Agar rost ifodaning qiymati noldan farqli o'zgarmas bo'lsa, cheksiz takrorlanish ro'y beradi. *Masalan*:

while(1); // cheksiz takrorlanish

Xuddi *for* operatoridagi kabi bu operatorda ham qavs ichida ',' yordamida ajratilgan bir nechta ifodalarni yozish mumkin.

Masala. Ikkita natural sonning eng katta umumiy bo'luvchisi (EKUB)ni *Evklid* algoritmi yordamida topish masalasi uchun algoritm va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char
*argv[])
{
    int a,b;
    /*Berilgan o'zgaruvchilarga
    qiymat kiritish*/
    cin>>a>>b;
    /*a va b lar uchun Ekub ni
    topish*/
    while (a!=b)
    { if (a>b) a=a-b;
      else b=b-a; }
    //Natijani chiqarish
    cout<<a;
    system("PAUSE");
    return EXIT_SUCCESS;}
  
```

Butun turdagi *a* va *b* qiymatlari oqimdan o'qilgandan keyin ular qiymatlari toki o'zaro teng bo'lmaguncha takrorlash jarayoni ro'y beradi. Takrorlashning har bir qadamida *a* va *b* sonlarining kattasidan, kichigi ayriladi va ularning tengligi tekshiriladi. Takrorlashdan keyingi

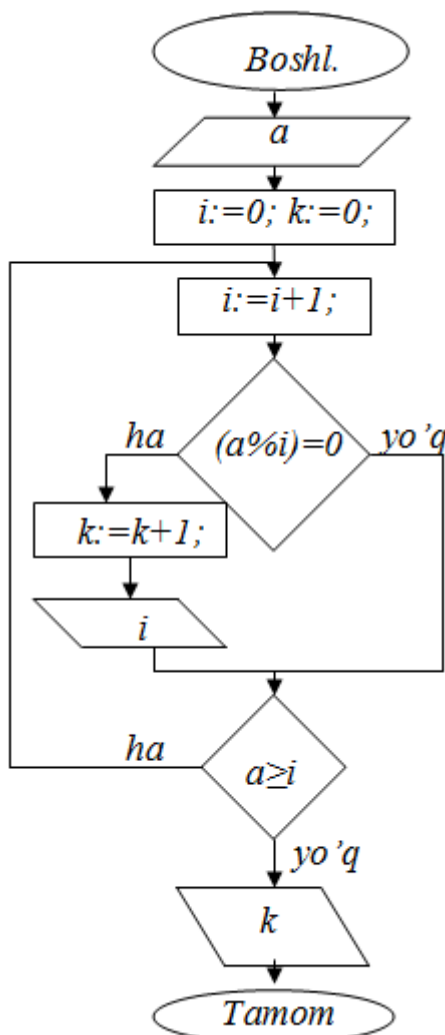
ko'rsatma vositasida a o'zgaruvchisining qiymati natija sifatida chop etiladi.

do-while takrorlash operatori *while* operatoridan farqli ravishda oldin operator yoki blokni bajaradi, keyin takrorlanish shartini tekshiradi. Bu qurilma takrorlanish tanasining kamida bir marta bajarilishini ta'minlaydi. *do-while* takrorlash operatori quyidagi sintaksisga ega:

do <operator yoki blok> *while* (<ifoda>);

Bunday takrorlash operatorining qo'llaniladigan holatlari takrorlanishni boshlamasdan turib, takrorlanish shartini tekshirishning iloji bo'lmagan holatlar hisoblanadi. *Masalan*, birorta jarayonni davom ettirish yoki to'xtatish haqidagi so'rovga javob olish va uni tekshirish zarur bo'lsin. Ko'rinib turibdiki, jarayonni boshlamasdan oldin bu so'rovni berishning ma'nosi yo'q. Hech bo'lmaganda takrorlanish jarayonining bitta qadami amalga oshirilgan bo'lishi kerak.

Masala. Berilgan a musbat butun sonining barcha musbat butun bo'luvchilari va ularning miqdorini chop etadigan algoritm va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
#include<stdio.h>
using namespace std;
int main(int argc, char
*argv[])
{ unsigned int a;
/*berilgan musbat sonni a
o'zgaruvchisiga o'qish*/
scanf("%d",&a);
/*bo'luvchilar soni va
bo'luvchini o'zida saqllovchi
o'zgaruvchilar*/
int k=0,i=0;
do { i++;
/*berilgan son
bo'luvchilarini topish,ularni
miqdorini hisoblash va chop
etish*/
if(!(a%i)){ k++;
cout<<i<<"\n";
cout<<"\n";}}
while(a>=i);
/*son bo'luvchilari miqdorini
chop etish*/
  
```

```

cout<<a<<" ning
bo'luvchilari soni "<<k<<" ta";
system("PAUSE");
return EXIT_SUCCESS;}

```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 3.061** Kesmalar uzunliklarini ifodalovchi a va b musbat sonlar berilgan ($a > b$). a uzunlikdagi kesmaga b uzunlikdagi kesma(lar)ni mumkin qadar eng ko'p miqdorda joylashtirganda, a kesmaning bo'sh (ortib) qolgan bo'lagi topilsin. Ko'paytirish va bo'lish amallaridan foydalanilmasin.
- 3.062** Kesmalar uzunliklarini ifodalovchi a va b musbat ($a > b$) sonlar berilgan. a uzunlikdagi kesmaga b uzunlikdagi kesma(lar) mumkin qadar eng ko'p miqdorda joylashtirilganda, (ko'paytirish va bo'lish amallaridan foydalanmay) a kesmaga joylashgan b kesmalar soni aniqlansin.
- 3.063** n va k musbat butun sonlari berilgan. Faqat qo'shish va ayirish amallaridan foydalanib, n ni k ga bo'lgandagi bo'linmaning butun hamda qoldiq qismlari aniqlansin.
- 3.064** n butun ($n > 0$) son berilgan. Agar u 3 sonining darajalaridan biri bo'lsa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.065** n butun ($n > 0$) son berilgan. U 2 sonining biror bir darajasidan iborat bo'lsa ($n = 2^k$), shu darajaning ko'rsatkichi k butun soni chiqarilsin.
- 3.066** n butun ($n > 0$) son berilgan. $n!!$ hisoblansin. Bu yerda $n!! = n(n-2)(n-4)...$ (oxirgi ko'paytuvchi agar n -juft bo'lsa 2 ga, toq bo'lsa 1 ga teng.) Natijaning butun tur diapozonidan oshib ketishining oldini olish uchun ko'paytma haqiqiy turga tegishli o'zgaruvchiga qiymatlanadi.
- 3.067** n butun ($n > 0$) son berilgan. Kvadratdan ildiz chiqarish formulasidan foydalanmay, kvadrati n dan katta bo'lgan dastlabki butun k soni topilsin. ($k^2 > n$)
- 3.068** n butun ($n > 0$) son berilgan. Kvadratdan ildiz chiqarish formulasidan foydalanmay kvadrati n dan katta bo'lmagan eng katta butun k soni topilsin. ($k^2 \leq n$)
- 3.069** n butun ($n > 1$) son berilgan. $3^k > n$ tengsizlik o'rinli bo'ladigan eng kichik k butun soni topilsin.

- 3.070** n butun ($n > 1$) son berilgan. $3^k < n$ tengsizlik o‘rinli bo‘ladigan eng katta k butun soni topilsin.
- 3.071** n butun ($n > 1$) son berilgan. $1 + 2 + \dots + k \geq n$ tengsizlikni qanoatlantiradigan eng kichik k butun soni va yig‘indining qiymati chiqarilsin.
- 3.072** n butun ($n > 1$) son berilgan. $1 + 2 + \dots + k \leq n$ tengsizlikni qanoatlantiradigan eng katta k butun son va yig‘indining qiymati chiqarilsin.
- 3.073** Bankdagi boshlang‘ich qo‘yilma n so‘m bo‘lsa va u har oyda p foiz ko‘payib borsa (p -haqiqiy son, $0 < p < 25$) necha oydan so‘ng qo‘yilma m so‘mdan oshishi hamda qo‘yilmaning oxirgi miqdori chiqarilsin.
- 3.074** Sportchi mashg‘ulotni boshladi. U 1-kun n km masofani bosib o‘tdi. Keyingi har kun bosib o‘tilgan yo‘l uzunligi oldingi kun bosib o‘tilgan yo‘ldan p foiz oshirildi (p haqiqiy son, $0 < p < 50$). Necha kundan so‘ng bir kunlik yugurish masofasi q km dan oshishi aniqlansin va jami bosib o‘tilgan masofa S hisoblansin.
- 3.075** n butun ($n > 1$) son berilgan. Butunga bo‘lish va qoldiqni aniqlash amallaridan foydalanib, berilgan sonning raqamlari alohida-alohida teskari tartibda chiqarilsin (birlik xonasidan boshlab).
- 3.076** n butun ($n > 1$) son berilgan. Butunga bo‘lish va qoldiqni aniqlash amallaridan foydalanib, uning raqamlari yig‘indisi va raqamlari soni chiqarilsin.
- 3.077** n butun ($n > 0$) son berilgan. Butunga bo‘lish va qoldiqni aniqlash amallaridan foydalanib, n sonini teskarisiga(o‘ngdan chapga) o‘qishdan hosil bo‘ladigan son chiqarilsin.
- 3.078** n butun ($n > 0$) son berilgan. Butunga bo‘lish va qoldiqni aniqlash amallaridan foydalanib n sonida 2 raqami borligi aniqlansin. Agar bor bo‘lsa 1 (rost) aks holda 0 (yolg‘on) chiqarilsin.
- 3.079** n butun ($n > 0$) son berilgan. Butunga bo‘lish va qoldiqni aniqlash amallaridan foydalanib n sonida toq raqam borligi aniqlansin. Agar bor bo‘lsa 1 (rost), aks holda 0 (yolg‘on) chiqarilsin.
- 3.080** Nol bo‘lmagan butun sonlar ketma ketligi berilgan. Uning tugaganlik simvoli(alomati) nol soni. Ketma ketlikdagi (noldan farqli) sonlar miqdori aniqlansin.
- 3.081** Nol bo‘lmagan butun sonlar ketma ketligi berilgan. Uning

tugaganlik simvoli(alomati) nol soni. Berilgan ketma ketlikdagi barcha musbat juft sonlar yig'indisi hisoblansin. Agar ketma ketlikda talab qilingan sonlar bo'lmasa nol chiqarilsin.

- 3.082** k butun son va nol bo'lmagan butun sonlar ketma ketligi berilgan. Uning tugaganlik simvoli nol soni. Ketma ketlikdagi k dan kichik, noldan farqli sonlar miqdori chiqarilsin.
- 3.083** k butun son va nol bo'lmagan butun sonlar ketma ketligi berilgan. Uning tugaganlik simvoli nol soni. Ketma ketlikdagi k dan katta birinchi uchragan son, agar bunday son bo'lmasa nol chiqarilsin.
- 3.084** k butun son va nol bo'lmagan butun sonlar ketma ketligi berilgan. Uning tugaganlik simvoli nol soni. Ketma ketlikdagi k dan katta oxirgi uchragan son, agar bunday son bo'lmasa nol chiqarilsin.

B toifa

- 3.085** n butun ($n > 1$) son berilgan. Agar u tub son bo'lsa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.086** a va b butun musbat sonlari berilgan. Evklid algoritmidan foydalanib ularning eng katta umumiy bo'luvchisi topilsin (EKUB). Agar $b \neq 0$ bo'lsa, $EKUB(a,b) = EKUB(b, a \% b)$, aks holda $EKUB(a,0) = a$.
- 3.087** n butun ($n > 0$) son berilgan. f_k Fibonachchi sonlar ketma-ketligi quyidagicha aniqlansa, $f_1 = 1, f_2 = 1, f_k = f_{k-2} + f_{k-1} \quad k = 3, 4, \dots$, n sonining Fibonachchi sonlar ketma-ketligida uchrashi tekshirilsin. Agar n soni uchrasa 1 (rost), aks holda 0 (yolg'on) chiqarilsin.
- 3.088** n butun ($n > 0$) son berilgan. f_k Fibonachchi sonlar ketma-ketligi $f_1 = 1, f_2 = 1, f_k = f_{k-2} + f_{k-1}$ uchun n dan katta birinchi uchragan Fibonachchi soni topilsin.
- 3.089** $n(n > 1)$ butun son berilgan. f_k Fibonachchi sonlar ketma-ketligi quyidagicha aniqlanadi. $f_1 = 1, f_2 = 1, f_k = f_{k-2} + f_{k-1} \quad k = 3, 4, \dots$
 $n = f_k$ bo'lsa f_{k+1} va f_{k-1} (oldingi va keyingi) Fibonachchi sonlari, aks holda 0 chiqarilsin.
- 3.090** n butun ($n > 0$) son berilgan. f_k Fibonachchi sonlar ketma-ketligi quyidagicha aniqlanadi. $f_1 = 1, f_2 = 1, f_k = f_{k-2} + f_{k-1} \quad k = 3, 4, \dots$
 $n = f_k$ bo'lsa, k (Fibonachchi sonining tartib nomeri), aks holda 0 chiqarilsin.
- 3.091** a, b, c musbat ($c \leq a, c \leq b$) sonlar berilgan. ab o'lchamli to'g'ri

to'rtburchakka tomoni c bo'lgan kvadratlar mumkin bo'lgan eng ko'p miqdorda joylashtirilganda, ko'paytirish va bo'lish amallaridan foydalanmasdan, to'g'ri to'rtburchakka joylashtirilgan kvadratlar soni aniqlansin.

- 3.092** k ta bo'sh bo'lmagan butun sonli ketma ketliklar berilgan, nol raqami har bir ketma ketlikning tugaganlik belgisi hisoblanadi (0 raqami ketma ketlik elementi hisoblanmaydi). Har bir ketma ketlikdagi elementlar soni hamda barcha ketma ketliklardagi jami elementlar soni chiqarilsin.
- 3.093** Nol bo'lmagan k ta sonli ketma ketliklar berilgan. Har bir ketma ketlik kamida 2 tadan elementni saqlaydi, nol raqami har bir ketma ketlikning tugaganlik belgisi hisoblanadi (0 raqami ketma ketlik elementi hisoblanmaydi). Elementlari o'sish tartibida joylashgan ketma ketliklar soni topilsin.
- 3.094** Nol bo'lmagan k ta ($k > 0$) sonli ketma ketliklar berilgan. Har bir ketma ketlik kamida 2 tadan elementni saqlaydi, nol raqami har bir ketma ketlikning tugaganlik belgisi hisoblanadi (0 raqami ketma ketlik elementi hisoblanmaydi). Elementlari o'sish yoki kamayish tartibida joylashgan (0 hisobga olinmaydi) ketma ketliklar soni topilsin.
- 3.095** Nol bo'lmagan k ta ($k > 0$) sonli ketma ketliklar berilgan, nol raqami har bir ketma ketlikning tugaganlik belgisi hisoblanadi (0 raqami ketma ketlik elementi hisoblanmaydi). Har bir ketma ketlik uchun quyidagicha harakatlar bajarilsin: Agar ketma ketlik elementlari o'suvchi bo'lsa 1 , kamayuvchi bo'lsa -1 , kamayuvchi ham o'suvchi ham bo'lmasa 0 chiqarilsin.
- 3.096** Nol bo'lmagan k ta ($k > 0$) sonli ketma ketliklar berilgan, nol raqami har bir ketma ketlikning tugaganlik belgisi hisoblanadi (0 raqami ketma ketlik elementi hisoblanmaydi). Har bir ketma ketlik kamida 3 tadan elementni saqlaydi. Elementlari arrasimon joylashgan ketma ketliklar soni topilsin.
- 3.097** Nol bo'lmagan k ta ($k > 0$) sonli ketma ketliklar berilgan. Har bir ketma ketlik kamida 3 tadan elementni saqlaydi va nol raqami har bir ketma ketlikning tugaganlik belgisi hisoblanadi (0 raqami ketma ketlik elementi hisoblanmaydi). Har bir ketma ketlik uchun quyidagicha harakatlar bajarilsin: agar ketma ketlik elementlari arrasimon bo'lsa uning elementlari soni, boshqa vaziyatlarda uning birinchi elementi chop etilsin.

3.3. Bobga doir qiyinroq masalalar

3.098

Uch o'lchovli rux[11].

Uch o'lchovli shaxmat o'yini $N \times N \times N$ o'lchovli kub shaklidagi maydonda o'ynalmoqda. Uch o'lchovli rux istalgan kataklarga to'g'ri chiziq bo'yicha, ya'ni 6 yo'nalish bo'yicha harakatlanishi mumkin (istalgan tomonga har bir 3 yo'nalish bo'yicha). Bunaqa maydonda K ta rux qo'yilgan. Ular butun maydonni ta'qib eta olishini yoki eta olmasligini aniqlovchi dastur tuzilsin.

Berilganlar:

Birinchi qatorda kubning o'lchami natural son $N(1 \leq N \leq 1000)$ shaklida va ruxlar soni $K(0 \leq K \leq 10^6)$ berilgan. So'ng ruxlar koordinatalarini bildiruvchi K ta 3 talik son (koordinatalar har bir yo'nalish uchun - natural sonlar 1 dan N gacha) kiritilgan.

Natija:

Agar ruxlar butun kubni ta'qib eta olishsa "YES" so'zini, va aksincha bo'lsa "NO" so'zini chiqarish kerak. Agar "NO" bo'ladigan bo'lsa, ikkinchi qatorda ruxlar ta'qib eta olmaydigan biror bir katakning koordinatalarini chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
2 2 1 1 2 2 2	YES
2 2 1 1 1 1 1 2	NO 2 2 1

3.099

EKUB[11].

Vali eng katta umumiy bo'luvchi (EKUB) ni topishni o'rgandi va bu o'rganilgan bilimni amaliyotga qo'llashga qaror qildi. U qog'ozga n ta son yozdi $a_1, \dots, a_n$ – bu sonlar uning o'rtoqlari yashovchi uylarning nomeri edi. Endi u shu sonlar ichidan shundaylarini topishi kerakki, bu sonlarning EKUBi u yoqtirgan d soniga teng bo'lishi kerak. Valiga yozilgan sonlar ichidan u qidirayotgan sonlar to'plamini topishga yordam bering.

Berilganlar:

Birinchi satrda ikkita n va d ($1 < n < 1000$, $1 < d < 109$) sonlari , ikkinchi satrda esa n ta butun son beriladi: $a_1, a_2, \dots, a_n$ ($1 < a_i < 109$).

Natija:

Agar izlanayotgan sonlar to'plami mavjud bo'lsa undagi sonlar miqdori k , aks holda -1 chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
4 3 6 8 12 9	3
3 3 2 4 8	-1

3.100**Tort [10].**

Alining tug'ilgan kuniga tomonlari n va m bo'lgan to'g'ri to'rtburchakli tort sovg'a qilishdi. Ali to'g'ri to'rtburchaklardan ko'ra kvadratni yaxshi ko'radi. U tortni ko'plab bir xil kvadrat bo'lakchalarga bo'lishga ahd qildi. Afsuski, uning ukasi Alisher undan oldin tortga yetib keldi va tortni parallel tomonlarini bir necha marta kesishga ulgurdi. Alini bundan jahli chiqdi. Alisher uni ovutmoqchi. Buning uchun u kesib olingan bo'laklardan mumkin bo'lgan o'lchamdagi maksimal kvadratlarni chiqarishi kerak. Boshida esa olingan bo'lakdan qanday eng katta o'lchamdagi kvadratni kesib olishini bilishi kerak. Alisherga yordam bering.

Berilganlar:

Birinchi satrda uchta n , t va k ($1 \leq n, m \leq 10^9$, $0 \leq k \leq 10^5$) butun sonlar berilgan. Keyingi k ta satrda bo'laklar berilgan. Har bir bo'lakda t va v sonlari berilgan. Dekart koordinalatalar sistemasini shunday kiritamizki, tortning bitta burchagi $(0,0)$, ikkinchi burchagi (n,m) bo'lsin. Demak, $t=0$ bo'lganda o'q bo'yicha bo'lak $x = v$ ($0 \leq x \leq n$) sohani o'z ichiga oladi, $t=1$ bo'lganda esa $y = v$ ($0 \leq y \leq m$) sohani.

Natija:

Tort bo'laklarga ajratilgandan keyin eng katta o'lchamdagi

tomonlari kvadrat bo'lgan bo'laklar soni chiqarilsin.

Berilganlar:	Natija:
10 10 2 1 5 0 3	5
4 7 0	4 6

3.101

Shavkat va baxtli bilet [12]

Shavkat juda tejamkor va hamisha u o'qishga avtobusda qatnaydi. Avtobusga minishdan yana bir maqsadi unga bilet olgan vaqtda uning baxtli bilet ekanligini tekshirib ko'rish. Bu uning sevimli mashg'uloti. Avtobus bileti $2n$ xonali son bo'lib (nol bilan boshlanishi mumkin), agar uning dastlabki n ta xonasi raqamlari yig'indisi oxirgi n ta xonasi raqamlari yig'indisiga teng bo'lsa, baxtli bilet deb aytiladi. Shavkat hamisha shu umid bilan avtobusga chiqadi, lekin hozirga qadar biror marta ham bunday bilet unga yo'liqmad. Bunday yurish joniga tekgach u baxtli bilet qanday bo'lishi kerakligini o'zi uchun o'zi o'ylab topdi. Uning fikricha: agar uning dastlabki n ta xonasi raqamlari yig'indisi oxirgi n ta xonasi raqamlari yig'indisidan katta bo'lsa, bilet baxtli bilet deb aytiladi. Bu qanchalik foydali ekanligini bilish uchun endi u $2n$ xonali sonlar ichida jami nechta baxtli bilet borligini hisoblab topmoqchi. Bu ish juda qiyin bo'lganli uchun unga yordam bering.

Berilganlar:

Birinchi qatorda bitta butun n soni beriladi ($1 \leq n \leq 9$).

Natija:

Birinchi qatorda Shavkatning nuqtai nazari bo'yicha $2n$ xonali baxtli biletlar sonini chiqaring.

Misollar:

Berilganlar:	Natija:
1	45

3.102

Elektropoyezd [11].

Elektropoyezdning ikkita qo'shni bekatlar orasidan o'tishda sarflaydigan vaqti va boshlang'ich bekatdan jo'nash vaqti ma'lum. Elektropoezdning har bir bekatdan jo'nash vaqtini hisoblovchi dastur tuzing (eng oxirgi bekat uchun kelish vaqti - poezdni bekatlarda turish vaqt hisobga olinmasin).

Berilganlar:

Birinchi qatorda poyezdning bosh bekatdan jo'nash vaqti berilgan. Vaqt quyidagi formatda beriladi: avval soatni bildiruvchi ikkita raqam (00 dan 23gacha), ikki nuqta va minutni bildiruvchi ikkita raqam (00dan 59gacha) berilgan. Vaqt kiritiladigan qatorda Bo'sh joy qo'yilishi mumkin emas. Ikkinchi qatorda elektropoyezd yo'nalishidagi bekatlar soni N ($2 \leq N \leq 1000$) (bosh va oxirgi bekatlar hisoblanadi) berilgan. Uchinchi qatorda $N-1$ ta son yozilgan – birinchi son - bosh bekatdan ikkinchi bekatga borguncha necha minut ketishini bildiradi, ikkinchi son – ikkinchi bekatdan uchinchigacha va hokazo. Har bir berilgan son natural son va 1000 dan katta emas.

Natija:

Har bir bekat uchun poezdni shu bekatdan o'tish vaqti chiqarilsin. Vaqtni kiritilgan formatda chiqarish lozim.

<i>Berilganlar:</i>	<i>Natija:</i>
07:00 4 10 5 3	07:00 07:10 07:15 07:18
22:58 5 2 60 43 20	22:58 23:00 00:00 00:43 01:03

3.103

Aksiya [10].

Bir kuni Vali magazinga keldi va “har bir k -chi tovar bepul” nomli aksiya bo'layotganini ko'rib qoldi. Vali aksiyani shartlari bilan tanishib chiqdi. Xaridor tovarlari bilan kassaga borib chek

oladi. Masalan chekda n ta tovar bo'lsa, u holda n/k ta tovarlar to'plamlarining eng arzonlari unga tekinga tushadi. Masalan, agar chekda 200, 100, 1000, 400 va 100 so'mli beshta tovar va $k = 2$ bo'lsa, u holda 100 so'mluk ikkita Tovar ham tekinga tushadi. Jami bo'lib xaridor 1600 so'm to'lashi zarur. Vali tovarlarni oldi va kassaga keldi. Tushundiki, agarda olgan tovarlarni bir nechta cheklarga bo'linsa kamroq pul to'lanar ekan. Valiga olgan tovarlarini bir nechta cheklarga bo'lgan holda kam mablag' to'lashiga yordam bering. Misoldagi Vali olgan tovarlarni ikkita chekka ajratamiz. Birinchi chekka 1000 va 400 so'mlisi, bunda 400 so'mlisi tekinga tushadi. Ikkinchi chekda esa 100 so'mli tovar tekinga tushadi. Demak jami bo'lib 1300 to'lanadi.

Berilganlar:

Birinchi satrda ikkita butun son berilgan n, k ($1 \leq n \leq 100\,000, 2 \leq k \leq 100$) – Vali sotib olgan tovarlar soni va “har bir k -chi tovar bepul” aksiyasini parametri. Keyingi satr Vali sotib olgan butun sonli n ta a_i ($1 \leq a_i \leq 10\,000$) tovar narxlari.

Natija:

Vali to'lagan *minimum* narx chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
5 2 200 100 1000 400 100	1300
6 4 100 11 56 22 43 62	251

3.104

Ketma-ketliklar.

Berilgan N soni uchun ketma-ket uchta bir raqami turmaydigan, nollar va birlardan tashkil topgan uzunligi N ga teng bo'lgan ketma-ketliklar soni aniqlansin.

Berilganlar:

35 dan oshmaydigan natural N soni berilgan.

Natija:

Izlanayotgan ketma-ketliklar soni chiqarilsin. Javoblar soni $2^{31}-1$

dan oshmasligi kafolatlanadi.

3.105

Spiral.

$N \times N$ katakli kvadratning kataklariga 1 dan N^2 gacha bo'lgan sonlar spiral shaklda yozilsin. ($2 \leq N \leq 100$).

Berilganlar:

Bitta satrda N soni berilgan.

Natija:

N ta satrning har birida N tadan son chiqarilsin. Spiralning uchi yuqori chap katakdan boshlansin. Spiral soat strekasi bo'yicha harakatlanadi.

3.106

Daraja.

Berilgan a va n natural sonlari uchun a^n ($1 \leq a \leq 9$, $1 \leq n \leq 7000$) hisoblansin.

Berilganlar:

Bitta satrda bo'sh joy bilan ajratilgan a va n sonlari berilgan.

Natija:

a^n chiqarilsin.

3.107

Og'irlik markazi.

Ko'pburchakni uchlarining koordinatalari boyicha uning og'irlik markazini koordinatasini toping. Ko'pburchakni tomonlari bir-biri bilan kesishmaydi (qo'shni uchlaridan tashqari). Ko'pburchakni maydoni nolga teng emas.

Berilganlar:

Birinchi satrda N ($3 \leq N \leq 100\,000$) soni berilgan. Keyingi N ta satrda ko'pburchak uchlarining koordinatalarini bildiruvchi sonlar juftligi berilgan. Agar berilgan tartib bo'yicha birinchi va oxirgi nuqtalarni birlashtirsak ko'pburchak hosil bo'ladi. Ko'pburchak uchlarining koordinatalari tekislikdagi dekart koordinatalar sistemasida berilgan va moduli bo'yicha 20 000 dan oshmaydi.

Natija:

Ko'pburchak og'irlik markazining koordinatasi 10^{-2} aniqlikda chiqarilsin.

3.108

Fibonachchi ketma ketligi.

$\{F_k\}$ – cheksiz sonlar ketma-ketligi berilgan, ular Fibonachchi $F_n = F_{k-1} + F_{k-2}$ shartini qanoatlantiradi (ixtiyoriy butun k uchun). i, F_i, j, F_j, n ($i \neq j$) berilgan. F_n ni toping. Ketma-ketlik qismiga misol:

k	-2	-1	0	1	2	3	4	5	6
F_k	-5	4	-1	3	2	5	7	12	19

Berilganlar:

Birinchi satrda i, F_i, j, F_j, n ($-1000 \leq i, j, n \leq 1000, -2\,000\,000 \leq F_k \leq 2\,000\,000\,000$ ($k = \min(i, j, n) \dots \max(i, j, n)$)) sonlari berilgan.

Natija:

F_n soni chiqarilsin.

3.109

Teng taqsimlash [11]

Uchta idishdan birida A milliliter (ml), ikkinchisida B ml va uchinchisida C ml suv bor. Quyidagi amallarni bajarish mumkin. Bir idishdan boshqa idishga suvni shunday quyishimiz kerakki, suv quyilayotgan idishdagi suvning hajmi quyilib bo'lgandan keyin oldingi hajmidan ikki barobar ko'p bo'lsin. Masalan, dastlab idishlarda A, B va C ml suv bor edi. Ikkinchi idishdan uchunchi idishga suv quyamiz. Quyib bo'lganimizdan so'ng idishlardagi suvning hajmi quyidagicha bo'ladi: $A, B-C, 2C$ ml (buni faqatgina $B \geq C$ bo'lgandagina amalgam oshirishimiz mumkin). Bu ishni ko'pi bilan 10000 marta takrorlashimiz mumkin. Ushbu ishni bajarish natijasida idishlardan birortasini bo'shata olishimizni aniqlab beruvchi dastur tuzilsin.

Berilganlar:

Idishlardagi suv miqdorini ifodalovchi manfiy bo'lmagan uchta A, B va C sonlari berilgan. A, B va C sonlar 10^{18} dan oshmaydi.

Natija:

Agar idishlardan birini bo'shatish mumkin bo'lsa, u holda uni bo'shatishga ketgan amallar soni va keyingi satrlarda bajarilgan amallar chiqarilsin. Har bir amal ikki sondan iborat, ya'ni quyayotgan va quyilayotgan idishlar raqami. Idishdan idishga quyish amallarini kamaytirish talan qilinmaydi, ammo undagi

amallar soni 10000 dan oshmasligi kerak. Agar idishni bo'shatishning imkoni bo'lmasa -1 chiqarilsin.

Berilganlar:	Natija:
1 2 10	2 3 1 2 1
0 1 0	0

3.110

Kutubxonachi-robot [11]

Bir kuni kutubxonachi-robot reviziya o'tkazishga qaror qildi va bitta kitob javoni ustidan Kormen kitobining o'ttiz uchinchi nashrining bir betini topib oldi. Bu betda qadimiy contest shartlari berilgan ekan. Robotga shartlarning formati aniq, lekin unga bu bet shubha uyg'otdi. Odatda har bir betning pasida «Страница i из n » struktura bo'lar edi, bunda i – varaq nomeri, n – esa barcha varaqlar sonini bildirardi. Lekin bu betda esa faqatgina bitta uzun raqamlar ketma-ketligi bor. Aftidan, nima uchundir printer raqamlardan boshqa simvollarni pechat qilmagan ekan. Shuning uchun i va n sonlari bir-biriga qo'shilib raqamlar ketma-ketligini hosil qilibdi.

Endi bu topilgan bet qaysi nomerda ekanligini aniqlash lozim. Buning yechimlari ko'p bo'lishi mumkin. Bunday yechimlar sonining nechtaligi robot uchun judayam qiziq. Lekin robotni o'zi bu muammoni hal qila olmaydi. Unga siz yordam bering. Varaqlar 1 dan n gacha nomerlanadi, hamda i va n sonlari oldidan hech qanaqa nollar qo'yilmaydi. «Страница i из n » ko'rinishidagi strukturalar soni nechtaligini aniqlashga yordam bering.

Malalan: 23507645 satr uchun 3 chiqishi lozim. Bular:

«Страница 2 из 3507645»

«Страница 23 из 507645»

«Страница 2350 из 7645»

Berilganlar:

Raqamlardan tashkil topgan satr berilgan. Satr uzunligi 1 dan 200000 gacha bo'lishi mumkin.

Natija:

«Страница i из n » ko'rinishidagi strukturalar soni chiqarilsin.

Berilganlar:	Natija:
23507645	3
1	0
1000000000	0

3.111

Choy [10]

Xalqaro olimpiadaning hakamlari choy ichishni juda ham yaxshi ko'rishadi. Lekin ular uchun hajmi v ml bo'lgan bitta choynak berilgan. Hakamlarni ish tartibi juda ham yaxshi shakllantirilgan. Shuning uchun, kim qachon choy ichishga borishi va uning chashkasiga qancha suv ketishi aniq. Ularning choy ichish jarayonini keltirib o'tamiz. Qachon hakamlardan biri choynak oldiga kelganda choynak ichidagi suvning hajmiga qaraydi. Ya'ni undagi suv chashkasi uchun yetarli? Agar choynakdagi suv hakam uchun yetarli bo'lmasa, u choynakka (v ml bo'lguncha) to'lguncha suv soladi. Suv solganidan so'ng choynakni yoqib toki suvni harorati 100 gradus bo'lguncha kutadi. Shundan so'ng o'z chashkasiga mos hajmdagi qaynoq suvni soladi va choynak yonidan ketadi.

Agar hakamlardan biri choynak oldiga kelgan vaqtda choynak qaynashini kutayotgan biror bir odam bo'lsa navbatga turadi va kutadi. Toki uning oldidagi odamlar ketmagunicha choynakka yaqinlashmaydi. Suvning qaynash haroratsi Selsiy boyicha 100 gradusni tashkil qiladi. Choynakning quvvati N bo'lib, agarda choynakda x ml suv bo'lsa uning haroratsi quyidagi fomula bo'yicha o'zgaradi: $T(t) = T_0 + Nt/x$, bunda T_0 - choynak yoqilayaotgan vaqtdagi suvning haroratsi bo'lsa, t - choynak yoqilgandan keyin o'tgan vaqt sekunlarda beriladi. Qachonki suvning harorati 100 gradusga yetsa choynak avtomatik ravishda o'chadi.

Choynak o'chirilgan vaqtdan boshlab undagi suvning harorati sovishni boshlaydi. Choynakdagi suvning hajmiga bo'g'liq

bo'lmagan holda uning harorati sekundiga k gradusga soviy boshlaydi. Qachonki, suvning harorati 20 gradus xona haroratiga teng bo'lganda sovish to'xtaydi. Choynakka solinayotgan suvning harorati 20 gradusga teng. Choynakka suv solganda uning temperaturasi mo'tadillashadi. Masalan, T haroratli suvga w ml suv salsak, suvning haroratsi $(wT + 20(v-w))/v$ ga teng bo'ladi.

Hakamlarning qachon choyga borishi va qancha qaynoq suv olishlari ma'lum. Har bir hakamning qachon choy ichish vaqtini aniqlash lozim. Hakamlar chashkaga qaynoq suv solishni shu lahzada amalga oshiradilar. Shuning uchun har doim ular o'zi uchun yetarli miqdordagi suvni chashkasiga solib oladilar. Dastlab choynak bo'sh bo'ladi.

Berilganlar:

Birinchi satrda to'rtta butun son berilgan. m, v, N va k ($1 \leq m \leq 100\,000$, $1 \leq v, N, k \leq 1000$) - mos ravishda choy ichadigan hakamlar soni, choynak hajmi ml da, choynakning quvvati va choynakdagi suvning sovish tezligi. Keyingi m ta satrda choy ichuvchilar berilgan. Har bir satrda ikkita butun son berilgan t_i va a_i ($1 \leq t_i \leq 10^6$; $1 \leq a_i \leq v$) – i -hakamning olimpiada boshlangan vaqtda hakam choy ichishga jo'nagan vaqt (sekundlarda) va unga kerak bo'ladigan suv hajmi, ya'ni chashkasining hajmi. Ikki va undan ortiq hakamlar bir vaqtda choy ichishga kelmaydi.

Natija:

Mos ravishda hakamlarni choy ichishni boshlagan vaqtlarini bildiruvchi sekundlar m ta satrda chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
3 1000 200 1 1 500 501 300 551 300	401.0 701.0 1021.0
1 1 1 1 1 1	81.0

Robot Bender o'zini tajribasini oshirish maqsadida attraksion ochishga qaror qildi. Attraksion quyidagilardan tashkil topgan: Bender 1 dan to k o'rinlarda joylashgan bir xil stakanchalardan birini tagiga sharikni yashiradi. Keyin n marta qandaydir stakanlar juftligini o'rnini almashtiradi. Shundan so'ng sharik qaysi stakan tagida joylashganligini topishni buyuradi. Bender robot bo'lganligi uchun ma'lum bir dastur asosida harakat qiladi. Bender x_i butun sonlar ketma-ketligini quradi, bunda $x_1=c$ va barcha $i > 1$ uchun $x_i = a * x_{i-1} + b$ ga teng. i -chi qadamda Bender stakanlarni $(x_i \% k) + 1$ va $((x_i + 1) \% k) + 1$ raqamlar boyicha almashtiradi. Boshida rabot sharikni n -raqamli stakanni tagiga yashiradi. Bender n ta almashtirishdan so'ng sharikni 1-raqamli stakan tagiga bo'lib qolishini xoxlaydi. Shunday a , b va c sonlarini topingki, sharik r -chi pozitsiyadan 1-chiga o'tib qolsin.

Berilganlar:

To'rtta son berilgan n , k , r va l ($1 \leq n \leq 10^5$; $2 \leq k \leq 10$; $1 \leq r, l \leq k$).

Natija:

Agarda bunday yechim mavjud bo'lmasa «Impossible» so'zi chiqarilsin. Aks holda manfiy bo'lmagan uchta a , b va c sonlari chiqarilsin. Sonlar 1000 dan katta emas.

<i>Berilganlar:</i>	<i>Natija:</i>
2 3 1 2	0 0 1
3 4 2 4	2 1
10 2 1 2	Impossible

3.113

Diafont tenglama [12].

$ax+by=c$ ko'rinishidagi tenglama berilgan. Sizning vazifangiz uning natural sonlardagi (x,y) yechimlarining sonini topishdan iborat.

Berilganlar:

Birinchi qatorda a , b va c butun sonlari berilgan ($1 \leq a, b, c \leq 10^6$).

Natija:

Berilgan tenglamaning yechimlari soni chiqarilsin.

<i>N_o</i>	<i>Berilganlar:</i>	<i>Natija:</i>
<i>1</i>	<i>2 3 25</i>	<i>4</i>
<i>2</i>	<i>5 7 11</i>	<i>0</i>

Izoh: 1-misolda (x,y) juftliklar: (2, 7), (5, 5), (8, 3), (11, 1).

IV Bob. Massivlar

Kompyuter xotirasida ketma-ket (regulyar) joylashgan bir xil turdagi qiymatlarga *massiv* deyiladi.

Katta hajmdagi cheklangan va tartiblangan qiymatlarni qayta ishlash bilan bog'liq masalalarni yechishda massivlardan foydalanishga zarurat tug'iladi.

Massivlarni matematikadagi sonlar vektoriga o'xshatish mumkin, chunki vektor ham o'zining individual nomiga ega va u chekli miqdordagi bir turdagi qiymatlardan, ya'ni sonlardan iborat.

Demak, *massiv* – bu chekli miqdordagi bir turdagi qiymatlar (*massiv* elementlari)ning tartiblangan majmuasidir. Ya'ni massivdagi barcha elementlar bir xil turga tegishli bo'lishi lozim. Bunday tur massiv elementining turi yoki massiv uchun tayanch tur deb ataladi. Dasturda ishlatiladigan har bir massiv o'zining individual nomiga ega bo'lishi kerak. Bu nom massiv o'zgaruvchisi deyiladi. Massivning har bir elementi massiv nomi, hamda kvadrat qavsga olingan va element selektori deb nomlanuvchi indeksni ko'rsatish orqali oshkor ravishda belgilanadi. Unga murojaat sintaksisi quyidagicha:

<massiv nomi>[<indeks>];

Bu ko'rinishga xususiy o'zgaruvchi deyiladi, chunki uning qiymati massivning alohida elementidir.

Umuman olganda indeks sifatida ifoda ishlatilishi mumkin. Ifoda qiymati massiv elementi nomerini aniqlaydi. Ifoda sifatida o'zgaruvchi ham olinishi mumkin, bunda o'zgaruvchining qiymati o'zgarishi bilan murojaat qilinayotgan massiv elementini aniqlovchi indeks ham o'zgaradi. Shunday qilib, dasturdagi bitta indeksli o'zgaruvchi orqali massivning barcha elementlarini aniqlash mumkin. Haqiqiy turdagi (*float*, *double*) qiymatlar to'plami cheksiz bo'lganligi sababli ular massiv indeksi sifatida ishlatilmaydi.

4.1. Dasturlashda bir o'lchovli massivlarni qo'llash

Bu bo'limda bir o'lchovli *statik* va *dinamik* massivlar bilan tanishamiz.

C++ tilida massiv indeksi doimo 0(nol) dan boshlanadi, uning eng katta qiymati massiv e'lonidagi uzunlikdan bittaga kam bo'ladi.

Massiv quyidagicha e'lon qilinadi:

$\langle \text{tur} \rangle \quad \langle \text{massiv} \quad \text{o'zgaruvchisi} \quad \text{nomi} \rangle \quad [\langle \text{o'lchami} \rangle]$
 $= \{ \langle \text{boshlang'ich qiymatlar} \rangle \};$

Bu yerda $\langle \text{o'lcham} \rangle$ - butun son ko'rinishidagi o'zgarmas ifoda. Bir o'lchovli massivlarni e'lon qilishga doir misollar:

$\text{int } a[5] = \{4, -5, 2, 10, 3\};$

$\text{float } n[4];$

Massiv *statik* yoki *dinamik* turda bo'lishi mumkin. *Statik* massivning uzunligi oldindan ma'lum bo'ladi va uning elementlari xotirada aniq bir adresdan boshlab ketma-ket joylashadi. *Dinamik* massivning uzunligi dastur bajarilishi jarayonida aniqlanadi va uning elementlari dinamik xotirada ayni paytda bo'sh bo'lgan adreslarga joylashadi. *Masalan*,

$\text{int } a[5];$ ko'rinishida e'lon qilingan bir o'lchovli massiv elementlari xotirada quyidagicha joylashadi:

Adres	Qiymatlar					
a	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]

4.1-jadval. Bir o'lchovli massivning xotiradagi joylashuvi.

Massivning i -elementiga $a[i]$ yoki $*(a+i)$ – vositali murojaat qilish mumkin. Massiv uzunligi $\text{sizeof}(a)$ amali orqali aniqlanadi.

Massiv e'lonida uning elementlariga boshlang'ich qiymatlar berish (initsializatsiyalash) mumkin va uning bir nechta variantlari mavjud.

1) O'lchami ko'rsatilgan massiv elementlarini to'liq initsializatsiyalash:

$\text{int } b[5] = \{8, 5, -11, 41, 39\};$

Bunda 5 ta elementdan iborat bo'lgan b nomli bir o'lchovli massiv e'lon qilingan va uning barcha elementlariga boshlang'ich qiymatlar berilgan. Bu e'lon quyidagi e'lon bilan ekvivalent:

$\text{int } b[5];$

$b[0] = 8; b[1] = 5; b[2] = -11; b[3] = 41; b[4] = 39;$

2) O'lchami ko'rsatilgan massiv elementlarini to'liqmas initsializatsiyalash:

$\text{int } b[5] = \{-11, 5, 29\};$

Bu yerda faqat massiv boshidagi uchta elementga boshlang'ich qiymatlar berilgan. Shuni aytib o'tish kerakki, massivni

initsializatsiyalashda uning boshidagi elementlariga boshlang'ich qiymatlar bermasdan turib, oxiridagi elementlariga boshlang'ich qiymatlar berish mumkin emas. Agarda massiv elementlariga boshlang'ich qiymat berilmasa, unda kelishuv bo'yicha *static* va *extern* modifikatori bilan e'lon qilingan massiv uchun elementlarning boshlang'ich qiymati *0(nol)* soniga teng bo'ladi. Avtomatik massiv elementlarining boshlang'ich qiymatlari esa noma'lum hisoblanadi.

3) O'lchami ko'rsatilmagan massiv elementlarini to'liq initsializatsiyalash:

```
int d[]={-15,7,15,24};
```

Bu misolda massivning barcha elementlariga qiymatlar berilgan hisoblanadi, massiv uzunligi esa kompilyator tomonidan boshlang'ich qiymatlar soniga qarab aniqlanadi.

Shuni ta'kidlash kerakki, massivning uzunligi berilmasa, unga boshlang'ich qiymatlar berilishi shart.

Bir o'lchovli statik massivlarni e'lon qilish:

```
char s[5]={ 'd', 'a', 'b', 'c', 'e' }; //belgilar massivi
```

```
int n[5] = {1,26,34,14}; // butun sonlar massivi
```

```
char str[]="defa"; /*sitr uzunligi 5 ga teng, chunki, uning oxiriga  
'\0' belgisi qo'shiladi.*/*
```

```
char str[]={ 'd', 'e', 'f', 'a' }; /*yuqoridagi satrning boshqacha  
yozilishi.*/*
```

Statik massivlarning kamchiliklari shundaki, ularning o'lchami oldindan ma'lum bo'lishi kerak, bundan tashqari bu o'lcham berilganlarga ajratilgan xotira segmentining o'lchami bilan chegaralangan. Ikkinchi tomondan, yetarlicha katta o'lchamdagi massiv e'lon qilib, masala yechilishida ajratilgan xotira to'liq ishlatilmasligi mumkin. Bu kamchiliklar dinamik massivlardan foydalanish orqali bartaraf etiladi, chunki ular dastur ishlashi jarayonida kerakli o'lchamdagi massivlarni yaratish va ularga zarurat qolmaganda yo'qotish imkoniyatini beradi.

Dinamik massivlarga xotiradan joy ajratish uchun *malloc()*, *calloc()* funksiyalaridan yoki *new* operatoridan foydalaniladi. Dinamik massivga ajratilgan xotirani bo'shatish uchun *free()* funksiyasi yoki *delete* operatori ishlatiladi.

Yuqorida qayd qilingan funksiyalar *alloc.h* kutubxonasida joylashgan.

malloc() funksiyasining sintaksisi

```
void * malloc(size_t ulcham) ;
```

ko‘rinishida bo‘lib, u xotiraning uyum qismidan *‘ulcham’* bayt o‘lchamidagi uzluksiz sohani ajratadi. Agar xotira ajratish muvaffaqiyatli bo‘lsa, *malloc()* funksiyasi shu soha boshlanishining adresini qaytaradi. Talab qilingan xotirani ajratish muvaffaqiyatsiz bo‘lsa, funksiya *NULL* qiymatini qaytaradi.

Sintaksisdan ko‘rinib turibdiki, funksiya *void* turidagi qiymat qaytaradi. Amalda esa aniq bir turdagi massiv ob‘ekti uchun xotiradan joy ajratish zarur bo‘ladi. Shu bois *void* turini aniq bir turga keltirish texnologiyasidan foydalaniladi. Masalan, butun turdagi uzunligi 3 ga teng massivga joy ajratishni quyidagicha amalga oshirish mumkin:

```
int * d_mas=(int*)malloc(3*sizeof(int));
```

malloc() funksiyasidan farqli ravishda *calloc()* funksiyasi massiv uchun joy ajratishdan tashqari massiv elementlarini 0(nol) qiymati bilan initsializatsiyalaydi. Bu funksiya sintaksisi quyidagi:

```
void * calloc (size_t miqdor, size_t ulcham) ;
```

ko‘rinishda bo‘lib, *‘miqdor’* parametri ajratilgan sohada nechta element borligini, *‘ulcham’* esa element o‘lchamini bildiradi.

free() dinamik xotirani bo‘shatish funksiyasi bo‘lib, ko‘rsatilgan dinamik massiv egallab turgan xotira qismini bo‘shatadi:

```
void free (void * blok);
```

free() funksiyasi parametrining *void* turida bo‘lishi, ixtiyoriy turdagi xotira bo‘lagini o‘chirish imkonini beradi.

Quyidagi dasturda 10 ta elementga ega dinamik massivni yaratish, unga dastlabki 10 ta toq sonni qiymatlash va massiv elementlari qiymatini chop etish hamda shu massivni xotiradan o‘chirish jarayonlari amalga oshirilgan.

```
#include <iostream.h>
#include <alloc.h>
int main()
{
    int * d_mas;
    if ((d_mas=(int*)malloc(10*sizeof(int)))==NULL)
    {
        cout<<"Xotira yetarli emas!!!";
        return 1;
    }
    // ajratilgan xotira sohasini to‘ldirish
    for(int i=0; i<10; i++) *(d_mas+i)=2*i+1;
    // dinamik massiv elementlarini chop etish
```

```

for(int i=0; i<10; i++) cout<<*(d_mas+i)<<"endl";
// ajratilgan xotira qismini bo'shatish
free (d_mas);
return 0;
}

```

Endi *new* va *delete* operatorlari bilan tanishamiz. *new* operatori yordamida massivga xotiradan joy ajratish uchun massiv turidan keyin kvadrat qavs ichida massiv elementlari soni ko'rsatiladi. Masalan, butun turdagi 10 ta sondan iborat massivga joy ajratish quyidagi

```
d_mas=new int[10];
```

ko'rinishda bo'ladi. Bu usulda massivga ajratilgan xotirani bo'shatish uchun

```
delete [] d_mas;
```

buyrug'ini berish kerak bo'ladi.

Massivlar bilan ishlashda quyidagilarni hisobga olish zarur:

- massiv elementlariga qiymat kiritish sikllar yordamida amalga oshirilishi maqsadga muvofiq;

- statik massivlar bilan ishlashda kiritiladigan ma'lumotlar soni, massiv elementlari sonidan oshmasligi kerak;

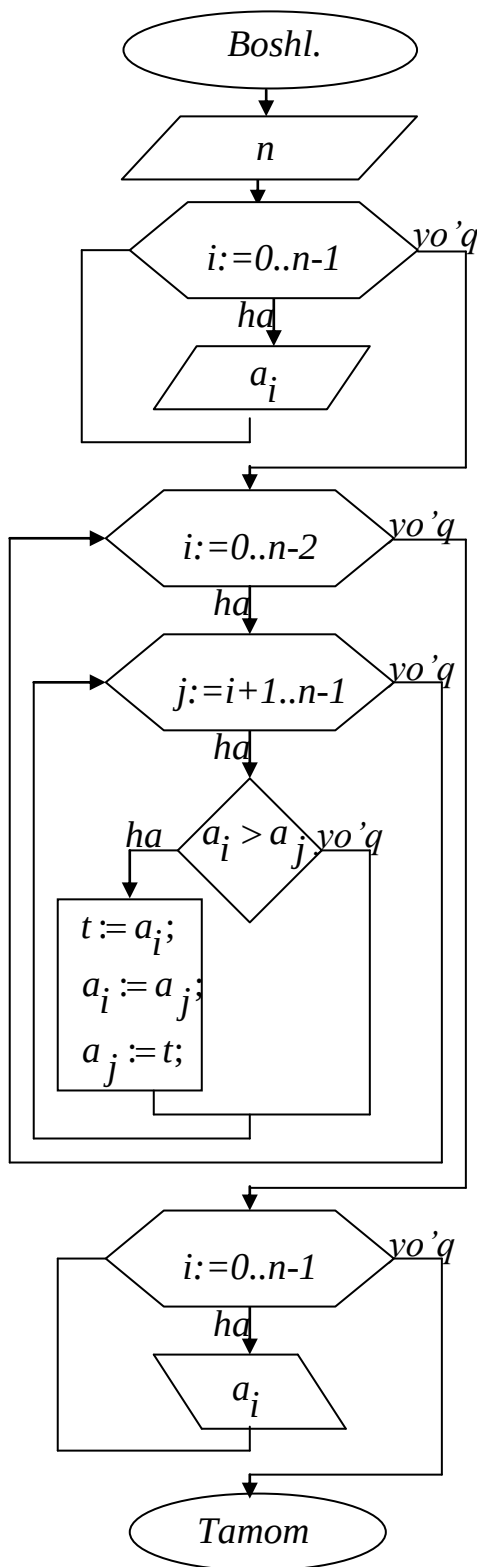
- massiv elementlaridagi barcha ma'lumotlarni chiqarish uchun ham sikllardan foydalanish maqsadga muvofiq;

Quyida keltiriladigan masalada bu tasdiqlar o'z aksini topgan. Shuni alohida ta'kidlash lozimki, *for* takrorlanish operatorini algoritmda tasvirlash uchun qo'llanmada ikki xil shakldan foydalanilgan, chunki C++ tilida *for* operatori murakkab konstruksiyali bo'lib, uning parametrlariga tur nuqtai-nazaridan cheklovlar qo'yilmaydi.

Qo'llanmaning 3-bobida takrorlanish operatorini tasvirlash uchun «*romb*» shakldan foydalanilgan. Quyida keltirilgan masalaning algoritmda esa takrorlanish operatorini tasvirlash uchun «*oltiburchak*» shakldan foydalanilgan. Chunki bu masalada barcha takrorlanish qadamlari 1 ga teng.

Yuqoridagilardan xulosa qiladigan bo'lsak, takrorlanishlar qadami 1 ga teng bo'lgan hol uchun algoritmlarni tasvirlashda «*oltiburchak*» shakldan, qolgan hollarda esa «*romb*» shakldan foydalanish maqsadga muvofiq bo'ladi.

Masala. n ta ($n > 0$) haqiqiy sondan tashkil topgan ketma-ketlik berilgan. Shu ketma-ketlikning elementlarini o'sish tartibida joylashtiruvchi algoritim va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
#include<stdio.h>
using namespace std;
int main(int argc, char
*argv[])
{ int n,i;
//massivga kiritiladigan
elementlar soni
printf("satr elementlari
sonini kiriting");
scanf("%d",&n);
/*haqiqiy son qabul qiluvchi
massiv e'lon qilish*/
float a[100];
/*massiv elementlariga
qiymatlar kiritish*/
for(int i=0;i<=n-1;i++)
cin>>a[i];
/*massiv elementlarini o'sish
tartibida tartiblash*/
for(int i=0;i<=n-2;i++)
for(int j=i+1;j<=n-1;j++)
if(a[i]>a[j]){
float t=a[i];
a[i]=a[j];
a[j]=t;}
/*tartiblangan
massiv elementlarini
chiqarish*/
for(int i=0;i<=n-1;i++)
cout<<a[i];
system("PAUSE");
return EXIT_SUCCESS;
}

```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 4.001** Dastlabki n ta ($n > 0$) musbat toq sonlarni saqlaydigan n o'lchamli butun sonli massiv tashkil etilsin.
- 4.002** 2 sonining darajalarini saqlaydigan n o'lchamli ($n > 0$) butun sonli massiv tashkil etilsin.
- 4.003** Arifmetik progressiyaning birinchi hadi a va uning ayirmasi d berilgan. Shulardan foydalanib o'zida arifmetik progressiyaning dastlabki n ta ($n > 1$) hadini saqlovchi massiv tashkil etilsin.
- 4.004** Geometrik progressiyaning birinchi hadi b va maxraji q berilgan. Shulardan foydalanib o'zida geometrik progressiyaning dastlabki n ta ($n > 0$) hadini saqlovchi massiv tashkil etilsin.
- 4.005** $f_1=1, f_2=1, f_k=f_{k-2}+f_{k-1}$ ($k=3,4,\dots$). f_k Fibonachchi sonlar ketma-ketligining dastlabki n ta ($n > 2$) hadini o'z ichiga oladigan n o'lchamli butun sonli massiv tashkil etilsin.
- 4.006** a va b butun sonlar berilgan. 1-elementi a ga, 2-elementi b ga, har bir keyingi elemeti barcha avvalgi elementlar (o'zidan oldingi barcha element) yig'indisiga teng bo'lgan n ($n > 2$) o'lchamli butun sonli massiv tashkil etilsin.
- 4.007** n o'lchamli ($n > 0$) a massiv berilgan. Uning elementlari teskari tartibda chiqarilsin.
- 4.008** n o'lchamli ($n > 0$) butun sonli massiv berilgan. Massivdagi juft sonlar va ularning miqdori k chiqarilsin.
- 4.009** n o'lchamli ($n > 1$) a massiv va $k(1 \leq k \leq n)$ butun soni berilgan. Massiv elementlari shart operatoridan foydalanmasdan quyidagi tartibda chop etilsin:
 $a_k, a_{k-1}, a_{k-2}, \dots, a_1$.
- 4.010** n o'lchamli ($n > 1$) a massiv berilgan (n - musbat juft son). Uning $a_0, a_2, a_4, \dots, a_{n-2}$ elementlari chiqarilsin.. Shart operatoridan foydalanilmasin.
- 4.011** n o'lchamli ($n > 0$) a massiv berilgan (n - musbat toq son). Uning $a_{n-1}, a_{n-3}, a_{n-5}, \dots, a_1$ chiqarilsin. Shart operatoridan foydalanilmasin.
- 4.012** n o'lchamli ($n > 1$) a massiv berilgan. Avval massivning $a_0, a_2, a_4, a_6, \dots$ elementlari, so'ngra $a_1, a_3, a_5, \dots$ elementlari chiqarilsin: Shart operatoridan foydalanilmasin.
- 4.013** n o'lchamli ($n > 1$) a massiv berilgan. Uning elementlari quyidagi tartibda chiqarilsin: $a_0, a_{n-1}, a_1, a_{n-2}, a_2, a_{n-3}, \dots$

- 4.014** n o'lchamli ($n > 3$, $n \% 4 = 0$) a massiv berilgan. Uning elementlari quyidagi tartibda chiqarilsin: $a_0, a_1, a_{n-1}, a_{n-2}, a_2, a_3, a_{n-3}, a_{n-4}, \dots$ (n -juft son).
- 4.015** n o'lchamli ($n > 0$) butun turga tegishli a massiv berilgan. Uning $a_k < a_{n-1}$ shartni ($0 \leq k \leq n-2$) qanoatlantiradigan birinchi a_k elementi chiqarilsin, bunday element topilmasa nol chiqarilsin.
- 4.016** n o'lchamli ($n > 2$) butun turga tegishli a massiv berilgan. Uning $a_0 < a_k < a_{n-1}$ shartni ($1 \leq k \leq n-2$) qanoatlantiradigan oxirgi a_k elementi chiqarilsin.
- 4.017** n o'lchamli ($n > 1$) massiv hamda k va l butun sonlari berilgan ($0 \leq k \leq l \leq n-1$). k -indeksdan l -indeksgacha bo'lgan (k va l ham kiradi) massiv elementlarining yig'indisi topilsin.
- 4.018** n o'lchamli ($n > 1$) massiv hamda k va l butun sonlari berilgan. ($0 \leq k \leq l \leq n-1$). k -indeksdan l -indeksgacha bo'lgan (k va l ham kiradi) massiv elementlarining o'rta arifmetigi topilsin.
- 4.019** Bir xil sonlarni o'z ichiga olmaydigan n o'lchamli ($n > 1$) butun turga tegishli massiv berilgan. Uning elementlari arifmetik progressiya hadlarini tashkil etishi aniqlansin. Agar tashkil etsa progressiya ayirmasi, aks holda 0 (nol) chiqarilsin.
- 4.020** Nol bo'lmagan butun sonli n ($n > 1$) o'lchamli massiv berilgan. Uning elementlari geometrik progressiya hadlarini tashkil etishi tekshirilsin. Agar tashkil etsa progressiya maxraji, aks holda 0 (nol) chiqarilsin.
- 4.021** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivda juft va toq sonlarning (navbat bilan) almashinib kelishi aniqlansin. Agar almashinib kelsa 0 , aks holda qonuniyatni buzgan birinchi element chiqarilsin.
- 4.022** n o'lchamli ($n > 0$) nol bo'lmagan butun turga tegishli massiv berilgan. Massivda musbat va manfiy sonlarning almashinib kelishi tekshirilsin. Agar almashinib kelsa 0 , aks holda qonuniyatni buzgan birinchi element chiqarilsin.
- 4.023** n o'lchamli ($n > 0$) a massiv berilgan. Uning $a_0, a_2, a_4, \dots$ elementlari ichidan eng kichigi topilsin.
- 4.024** n o'lchamli ($n > 0$) a massiv berilgan. Uning $a_1, a_3, a_5, \dots$ elementlari ichidan eng kattasi topilsin.
- 4.025** n o'lchamli ($n > 0$) massiv berilgan. O'zining o'ng yonidagi elementdan katta massiv elementlari indekslari va shunday elementlar soni, agar bunday element bo'lmasa 0 chiqarilsin.
- 4.026** n o'lchamli ($n > 0$) massiv berilgan. O'zining chap yonidagi

elementdan katta bo'lgan massiv elementlarining indeksleri va shunday elementlar soni, agar bunday element bo'lmasa 0 chiqarilsin.)

- 4.027** n o'lchamli ($n > 1$) massiv berilgan. Uning birinchi lokal minimumining indeksi, bunday element bo'lmasa 0 chiqarilsin (*lokal minimum – o'zining har ikki yonidagi elementdan kichik bo'lgan element. Massivdagi birinchi va oxirgi element lokal minimum bo'la olmaydi*).
- 4.028** n o'lchamli ($n > 1$) massiv berilgan. Uning oxirgi lokal maksimumining indeksi, bunday element bo'lmasa 0 chiqarilsin (*lokal maksimum – o'zining har ikki yonidagi elementdan katta bo'lgan element. Massivdagi birinchi va oxirgi element lokal maksimum bo'la olmaydi*).
- 4.029** n o'lchamli ($n > 1$) massiv berilgan. Uning lokal minimumlari orasidan eng kattasi, bunday element bo'lmasa 0 chiqarilsin.
- 4.030** n o'lchamli ($n > 1$) massiv berilgan. Uning lokal maksimumlari orasidan eng kichigi, bunday element bo'lmasa 0 chiqarilsin.
- 4.031** n o'lchamli ($n > 1$) massiv berilgan. Uning lokal maksimumini ham lokal minimumini ham tashkil etmaydigan eng katta element, bunday element bo'lmasa 0 chiqarilsin.
- 4.032** r soni va n o'lchamli ($n > 0$) a massiv berilgan. r soniga eng yaqin bo'lgan massiv elementi topilsin, agar bunday elementlar soni bir nechta bo'lsa, ulardan ixtiyoriy biri chiqarilsin. (a_k element r soniga yaqin element deyiladi, agar $|a_k - r|$ qiymat minimal bo'lsa, $0 \leq k \leq n-1$).
- 4.033** n o'lchamli ($n > 1$) massiv berilgan. Massivdagi yig'indisi eng katta bo'ladigan 2 ta yonma-yon turuvchi elementlar topilib, bu elementlarning indeksleri chiqarilsin. Bunday yig'indilar bir nechta bo'lsa oxirgisi olinsin.
- 4.034** r soni va n o'lchamli ($n > 0$) massiv berilgan. Yig'indisi r soniga eng yaqin bo'lgan 2 ta yonma-yon massiv elementlari topilib, bu elementlarning indeksleri o'sish tartibida chiqarilsin. Bunday yig'indilar bir nechta bo'lsa oxirgisi olinsin.
- 4.035** n o'lchamli ($n > 0$) massiv berilgan. Massivning ikkita eng yaqin (elementlar ayirmasi moduli eng kichik bo'lgan) elementlari, bunday elementlar juftligi bir nechta bo'lsa oxirgi uchragani chiqarilsin.
- 4.036** r soni va n o'lchamli ($n > 1$) massiv berilgan. Yig'idisi r ga eng yaqin bo'lgan 2 ta element, bunday elementlar juftligi bir

nechta bo'lsa oxirgi uchragani chiqarilsin.

4.037 Bir xil n o'lchamli ($n>0$) a va b massivlar berilgan. a va b massivlardagi mos elementlarning qiymatlari almashtirilsin. Avval a massiv elementlari keyin b massiv elementlari chiqarilsin.

4.038 n o'lchamli ($n>0$) a massiv berilgan. Xuddi shunday o'lchamli elementlari quyidagi ko'rinishda aniqlanadigan yangi b massiv hosil qilinsin.

$$b_k = 2a_k \text{ agar } a_k < 5, \text{ aks holda } b_k = \frac{a_k}{2}. 0 \leq k \leq n-1.$$

4.039 Bir xil n o'lchamli ($n>0$) 2 ta a va b massivlar berilgan. Shunday c massiv tuzilsin: c massivning i -indeksdagi elementi a va b massivlarning i -indeksdagi elementlarining kattasidan iborat bo'lsin.

4.040 n o'lchamli ($n>0$) butun sonli a massiv berilgan. Massivdagi barcha juft sonlar yangi butun turga tegishli b massivga yozilsin va hosil qilingan b massivning o'lchami hamda uning elementlari chiqarilsin.

4.041 n o'lchamli ($n>0$) butun turga tegishli a massiv berilgan. a massivning $a_1, a_3, a_5, \dots$ elementlarini yangi butun turga tegishli b massivga yozib, uning o'lchami hamda elementlari chiqarilsin. Shart operatoridan foydalanilmasin.

4.042 n o'lchamli ($n>1$) butun turga tegishli a massiv berilgan. Shunday o'lchamli yangi butun turga tegishli b massivga a massivning elementlari quyidagi tartibda yozilsin: $a_0, a_2, \dots, a_{n-2}, a_1, a_3, \dots, a_{n-1}$. Shart operatoridan foydalanilmasin.

4.043 n o'lchamli ($n>0$) a massiv berilgan. Quyidagi tartib bo'yicha shunday o'lchamli yangi b massiv tashkil etilsin: b_k elementi ($0 \leq k \leq n-1$) a massivning 0-dan k gacha indeksli elementlar yig'indisiga teng (k ham kiradi).

4.044 n o'lchamli ($n>0$) a massiv berilgan. Quyidagi qoida bo'yicha shunday o'lchamli yangi b massiv tuzilsin: b_k elementi ($0 \leq k \leq n-1$) a massivning 0-dan k -gacha indeksli elementlarining o'rta arifmetigiga teng. (k ham kiradi.)

4.045 n o'lchamli ($n>0$) a massiv berilgan. Shunday o'lchamli yangi b massiv quyidagi qoida bo'yicha tashkil etilsin: b_k elementi ($0 \leq k \leq n-1$) a massivning k -dan $n-1$ gacha indeksli elementlari yig'indisiga teng.

4.046 n o'lchamli ($n>0$) a massiv berilgan. Shunday o'lchamli yangi

- b massiv quyidagi tartib(qoida) bo'yicha tashkil etilsin: b_k elementi ($0 \leq k \leq n-1$) a massivning k -dan $n-1$ gacha indeksli elementlari o'rta arifmetigiga teng.
- 4.047** n o'lchamli ($n > 0$) a massiv berilgan. 2 ta yangi b va c massivlar hosil qilinsin: b massivga a massivning barcha musbat elementlari, c massivga manfiy elementlari yozilib, avval b massivning o'lchami va tarkibi, keyin c massivning o'lchami va tarkibi chiqarilsin.
- 4.048** n o'lchamli ($n > 0$) elementlari o'sish tartibida tartiblangan a va b massivlar berilgan. a va b massivlardagi barcha elementlarni o'z ichiga olgan, o'sish tartibidagi c massiv hosil qilinsin.
- 4.049** Elementlari kamayish bo'yicha tartiblangan 3 ta butun turga tegishli mos ravishda n_a , n_b , n_c o'lchamli a , b va c massivlar berilgan. Bu massivlar $n_a + n_b + n_c$ o'lchamli d massivga kamayish bo'yicha tartiblab birlashtirilsin.
- 4.050** n o'lchamli ($n > 0$) a massiv va k butun son ($0 \leq k \leq n-1$) berilgan. a_k elementning dastlabki qiymati uning har bir elementiga ko'paytirilsin.
- 4.051** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivda uchragan birinchi juft sonning dastlabki qiymati massivdagi barcha juft sonlarga ko'paytirilsin. Agar massivda juft son bo'lmasa, massiv o'zgartirishsiz qoldirilsin.
- 4.052** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivdagi oxirgi uchragan toq sonning dastlabki qiymati massivdagi barcha toq sonlarga ko'paytirilsin. Agar massivda toq son bo'lmasa, massiv o'zgartirishsiz qoldirilsin.
- 4.053** n o'lchamli ($n > 0$) massiv berilgan. Uning eng katta va eng kichik elementlari o'zni almashtirilsin.
- 4.054** n o'lchamli ($n > 0$) massiv berilgan (n -juft son). Uning a_0 elementini a_1 bilan, a_2 elementini a_3 bilan va hokazo. Xuddi shu tarzda elementlar o'zni almashtirilsin.
- 4.055** n o'lchamli ($n > 0$) massiv berilgan (n -juft son). Massivning birinchi yarmi bilan ikkinchi yarmining o'zni almashtirilsin.
- 4.056** n o'lchamli a massiv hamda k va l butun sonlar berilgan ($0 \leq k < l \leq n-1$). a_k va a_l elementlar orasida joylashgan massiv elementlari o'sish tartibda tartiblanib, hosil bo'lgan yangi massiv elementlari chiqarilsin. a_k va a_l ham kiradi.
- 4.057** n o'lchamli ($n > 1$) massiv berilgan. Eng kichik va eng katta elementlari orasida joylashgan massiv elementlari 0 ga

- aylantirilsin(*min* va *max* elementlar kirmaydi).
- 4.058** n o'lchamli ($n > 1$) massiv berilgan. Uning hamma lokal maksimumlari 0 bilan almashtirilsin. Agar bunday elementlar topilmasa, massiv o'zgarishsiz qoldirilsin.
- 4.059** n o'lchamli ($n > 1$) massiv berilgan. Uning hamma lokal minimumlari kvadratga ko'tarilsin. Agar bunday elementlar topilmasa, massiv o'zgarishsiz qoldirilsin.
- 4.060** n o'lchamli ($n > 0$) massiv berilgan. Massiv elementlarini bir qadam o'ngga siljitish amalga oshirilsin. (bunda $a_0 \rightarrow a_1$ ga o'tadi, $a_1 \rightarrow a_2$, ..., $a_{n-2} \rightarrow a_{n-1}$ ga o'tadi.) Massivning $n-1$ -elementining oldingi qiymati yuqoladi, 0 -elementining qiymati 0 ga teng bo'lib qoladi.
- 4.061** n o'lchamli ($n > 1$) a massiv berilgan. Massivning qo'shni elementlarini taqqoslash(a_0 va a_1 , a_1 va a_2 va hokazo) va agar juftlikdagi chap elementdan o'ng elementi katta bo'lsa, ularning joylarini almashtirish yo'li bilan bu harakatni $n-1$ marta takrorlab, massivning elementlarini oddiy almashtirish usuli(pufakli saralash) bilan massiv o'sish tartibida tartiblansin.
- 4.062** n o'lchamli ($n > 1$) a massiv berilgan. Massiv oddiy tanlash usuli yordamida tartiblansin: "undagi eng katta element topiladi, u massivning n -elementi bilan almashtiriladi. Bu holat qolgan $n-1$ ta element uchun yana takrorlanadi: Bu harakat jami $n-1$ marta bajariladi".
- 4.063** n o'lchamli ($n > 0$) butun turga tegishli a massiv berilgan. 2 ta bir xil o'lchamli b va c massivlar quyidagicha hosil qilinsin: b massivga berilgan massivdagi sonlar seriyalari uzunligi, c massivga esa seriyani tasvirlovchi sonlar yozilsin. (Seriya deganda bir xil sonlarni ketma-ketligi tushuniladi, seriya uzunligi deganda (u 1 ga ham teng bo'lishi mumkin) ketma-ket keluvchi bir xil sonlarning miqdori tushuniladi).
- 4.064** n o'lchamli ($n > 0$) butun turga tegishli a massiv berilgan. Shu massivdagi har bir sonlar seriyasidan oldin 0 elementi joylashtirilsin.
- 4.065** n o'lchamli ($n > 0$) butun turga tegishli a massiv berilgan. Shu massivdagi har bir sonlar seriyasidan keyin 0 elementi joylashtirilsin.
- 4.066** n o'lchamli ($n > 0$) butun turga tegishli a massiv berilgan. Massivning har bir seriyasiga bittadan mos element qo'shilsin.
- 4.067** n o'lchamli ($n > 0$) butun turga tegishli a massiv berilgan. Agar

- massivda uzunligi 1 dan katta sonlar seriyasi uchrasa, shu seriya uzunligi 1 birlik kamaytirilsin.
- 4.068** k butun ($k > 0$) soni va butun turga tegishli n o'lchamli ($n > 0$) massiv berilgan. Massivning k -seriyasi ikkilantirilsin. Agar massivdagi seriyalar soni k dan kichik bo'lsa, u o'zgarishsiz chop etilsin.
- 4.069** k butun ($k > 0$) soni va butun turga tegishli n o'lchamli ($n > 0$) massiv berilgan. Massivning k -seriyasi chiqarilsin. Agar massivdagi seriyalar soni k dan kichik bo'lsa, massivning o'zi chop etilsin.
- 4.070** l butun soni va butun turga tegishli n o'lchamli ($0 < l < n$) massiv berilgan. Massivdagi l dan kichik uzunlikka ega bo'lgan sonlar seriyalari 0 qiymati bilan almashtirilsin.
- 4.071** l butun soni va butun turga tegishli n o'lchamli ($0 < l < n$) massiv berilgan. Massivdagi l uzunlikka ega bo'lgan sonlar seriyalari 0 qiymati bilan almashtirilsin.
- 4.072** l butun soni va butun turga tegishli n o'lchamli ($0 < l < n$) massiv berilgan. Massivdagi l dan katta uzunlikka ega bo'lgan sonlar seriyalari 0 qiymati bilan almashtirilsin.
- 4.073** n o'lchamli ($n > 1$) butun turga tegishli massiv berilgan. Massivning birinchi seriyasining uzunligi birdan katta bo'lsa uning barcha seriyalarining uzunligi 1 ga oshirib tasvirlansin, aks holda massiv o'zgartirishsiz qoldirilsin.
- 4.074** n o'lchamli ($n > 1$) butun turga tegishli massiv berilgan. Massivdagi oxirgi seriyaning uzunligi 1 ga oshirilib, so'ngra natijaviy massiv chop etilsin.
- 4.075** Tekislikda n ta ($n > 0$) nuqtadan iborat A to'plam va B nuqta (nuqtalar o'zlarining x va y koordinatalari bilan) berilgan. A to'plamdan B nuqtaga eng yaqin bo'lgan nuqta topilsin. (Nuqtalar orasidagi masofa r quyidagi formula bilan aniqlanadi: $r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$)
- 4.076** Tekislikda n ($n > 0$) ta nuqtadan iborat A to'plam (nuqtalar x va y koordinatalari bilan) berilgan. To'plam nuqtalari orasidan 2-chorakda yotuvchi va koordinata boshidan eng uzoqda joylashgan nuqta koordinatalari, agar bunday nuqta bo'lmasa, nol chiqarilsin.
- 4.077** Tekislikda n ($n > 0$) ta nuqtadan iborat A to'plam (nuqtalar x va y koordinatalari bilan) berilgan. To'plam nuqtalari orasidan 1-

yoki 3- chorakda yotuvchi hamda koordinata boshiga eng yaqin bo'lgan nuqta koordinatalari, agar bunday nuqta bo'lmasa, nol chiqarilsin.

B toifa

- 4.078** n o'lchamli ($n > 1$) massiv berilgan. Uning monoton o'suvchi bo'laklari soni topilsin.
- 4.079** n o'lchamli ($n > 1$) massiv berilgan. Uning monoton kamayuvchi bo'laklari soni topilsin.
- 4.080** n o'lchamli ($n > 1$) massiv berilgan. Uning monoton oraliqlar (ya'ni elementlar o'suvchi yoki kamayuvchi bo'lgan bo'laklar) soni topilsin.
- 4.081** Kamida ikkita bir xil elementga ega bo'lgan n o'lchamli ($n > 1$) butun turga tegishli massiv berilgan. Bir xil elementlarning indeksleri chiqarilsin.
- 4.082** n o'lchamli butun sonli massiv berilgan. Agar u o'rin almashtirishlardan iborat bo'lsa ya'ni 1 dan n gacha bo'lgan barcha butun sonlarni o'z ichiga olsa 0 chiqarilsin aks holda 1-qonuniyatni buzadigan element chiqarilsin.
- 4.083** Tekislikda n ($n > 0$) ta nuqtadan iborat A to'plam (nuqtalar x va y koordinatalari bilan) berilgan. To'plamdagi bir-biridan eng uzoq masofada joylashgan nuqtalar juftligi topilsin.
- 4.084** Tekislikda mos ravishda n_1 va n_2 ta ($n_1 > 0$, $n_2 > 0$) nuqtalardan iborat A va B nuqtalar to'plami (nuqtalar x va y koordinatalari bilan) berilgan. Har bir to'plamdan bittadan nuqta topilsinki, bu nuqtalar orasidagi masofa eng qisqa bo'lsin.
- 4.085** Tekislikda n ta ($n > 2$) nuqtadan iborat A to'plam (nuqtalar x va y koordinatalari bilan) berilgan. To'plamning 3 ta har xil nuqtasida uchlari joylashgan perimetri eng katta bo'lgan uchburchak uchlari koordinatalari topilsin.
- 4.086** n ta nuqtaga ega A to'plam (nuqtalar x va y koordinatalari bilan) berilgan. To'plamdagi nuqtalar quyidagi shart asosida o'sish tartibida joylashtirilsin: $x_1 \leq x_2$ va $y_1 \leq y_2$ bo'lsa $(x_1, y_1) < (x_2, y_2)$.
- 4.087** n ta nuqtaga ega A to'plam berilgan. To'plamdagi nuqtalar quyidagi shart asosida o'sish tartibida joylashtirilsin: $x_1 + y_1 \leq x_2 + y_2$ va $x_1 < x_2$ bo'lsa $(x_1, y_1) < (x_2, y_2)$.
- 4.088** n o'lchamli ($n > 0$) massiv berilgan. Massiv elementlarini bir qadam chapga siljitish amalga oshirilsin (bunda $a_{n-1} \rightarrow a_{n-2}$ ga o'tadi, $a_{n-2} \rightarrow a_{n-3}$ ga..., $a_1 \rightarrow a_0$). Massivning 0-elementining

oldingi qiymati yuqoladi. $n-1$ - elementining qiymati 0 ga teng bo'lib qoladi.

- 4.089** n o'lchamli ($n > 0$) massiv va k butun son berilgan ($0 \leq k \leq n-1$). Massiv elementlarini k qadam o'ngga siljitish amalga oshirilsin (bunda $a_0 \rightarrow a_k, a_1 \rightarrow a_{k+1}$ ga..., $a_{n-k-1} \rightarrow a_{n-1}$ ga o'tadi). Massivdagi oxirgi k ta elementlarning qiymatlari yo'qoladi, dastlabki k ta elementlarning qiymatlari 0 ga teng bo'lib qoladi.
- 4.090** n o'lchamli ($n > 0$) massiv va k butun son berilgan. ($0 \leq k \leq n-1$). Massiv elementlarini k qadam chapga siljitish amalga oshirilsin. (bunda $a_{n-1} \rightarrow a_{n-k-1}, a_{n-2} \rightarrow a_{n-k-2}, \dots, a_k \rightarrow a_0$, ga o'tadi, massivning dastlabki k ta elementlarining qiymatlari yo'qoladi, oxirgi k ta elementlarining qiymatlari 0 ga teng bo'lib qoladi).
- 4.091** n o'lchamli ($n > 0$) massiv berilgan. Massiv elementlarini bir qadam o'ngga siklli siljitish amalga oshirilsin (bunda $a_0 \rightarrow a_1, a_1 \rightarrow a_2, \dots, a_{n-1} \rightarrow a_0$ ga o'tadi).
- 4.092** n o'lchamli ($n > 0$) massiv berilgan. Massiv elementlarini bir qadam chapga siklli siljitish amalga oshirilsin (bunda $a_{n-1} \rightarrow a_{n-2}, a_{n-2} \rightarrow a_{n-3}, \dots, a_0 \rightarrow a_{n-1}$ ga o'tadi).
- 4.093** n o'lchamli a massiv va k butun soni berilgan ($1 \leq k \leq 4, k < n$). Massiv elementlarini k qadam o'ngga siklli siljitish amalga oshirilsin. (bunda $a_0 \rightarrow a_k, a_1 \rightarrow a_{k+1}, \dots, a_{n-1} \rightarrow a_{k-1}$ ga o'tadi). 4 elementdan iborat yordamchi massivdan foydalanishga ruxsat etiladi.
- 4.094** n o'lchamli a massiv va k butun soni berilgan ($1 \leq k \leq 4, k < n$). Massiv elementlarini k qadam chapga siklli siljitish amalga oshirilsin (bunda $a_{n-1} \rightarrow a_{n-k-1}, \dots, a_0 \rightarrow a_{n-k}$) 4 elementdan iborat yordamchi massivdan foydalanishga ruxsat etiladi.
- 4.095** Bitta elementidan tashqari barcha elementlari kamayish tartibida tartiblangan n o'lchamli ($n > 1$) massiv berilgan. Tartiblanishni buzayotgan elementni tegishli o'ringa siljitib massiv to'liq tartiblansin.
- 4.096** n o'lchamli massiv va k butun son berilgan ($0 \leq k < n$). Massivning k -indeksdagi elementi o'chirilsin.
- 4.097** n o'lchamli massiv va k, l butun sonlari berilgan ($0 \leq k < l < n$). Massivning k -indeksdan l -indeksgacha bo'lgan (k va l ham kiradi) elementlari o'chirilsin va hosil bo'lgan massivning o'lchami hamda uning elementlari chiqarilsin.
- 4.098** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivdagi barcha toq sonlar o'chirilsin va hosil bo'lgan

- massivning o'lchami hamda elementlari chiqarilsin.
- 4.099** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivning $a_0, a_2, a_4, \dots, a_{n-2}$ elementlari o'chirilsin. Shart operatoridan foydalanilmasin.
- 4.100** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivning $a_1, a_3, \dots, a_{n-1}$ elementlari o'chirilsin. Shart operatoridan foydalanilmasin.
- 4.101** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. O'zaro teng bo'lgan qo'shni elementlarning birinchisi qoldirilib qolgan(lar)i o'chirilsin.
- 4.102** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivdagi o'zaro teng bo'lgan elementlarning birinchisi qoldirilib qolgan(lar)i o'chirilsin.
- 4.103** n o'lchamli ($n > 0$) butun turga tegishli massiv berilgan. Massivdagi o'zaro teng bo'lgan elementlarning oxirigisi qoldirilib, qolgan(lar)i o'chirilsin.
- 4.104** n o'lchamli ($n > 2$) butun turga tegishli massiv berilgan. Massivdagi 3 martadan kam uchraydigan elementlar o'chirilsin va hosil bo'lgan massivning o'lchami va tarkibi chiqarilsin.
- 4.105** n o'lchamli ($n > 2$) butun turga tegishli massiv berilgan. Massivdagi 2 martadan ko'p uchraydigan elementlari o'chirilsin. Hosil bo'lgan massivning o'lchami va tarkibi chiqarilsin.
- 4.106** n o'lchamli ($n > 1$) butun turga tegishli massiv berilgan. Massivdagi 2 marta uchraydigan elementlari o'chirilsin. Hosil bo'lgan massivning o'lchami va tarkibi chiqarilsin.
- 4.107** n o'lchamli massiv va k butun son berilgan ($0 \leq k < n$). Massivning k -indeksli elementining oldiga yangi 0 qiymatli element joylashtirilsin.
- 4.108** n o'lchamli massiv va k butun son berilgan ($0 \leq k < n$). Massivning k -indeksli elementidan keyin 0 qiymatli element joylashtirilsin.
- 4.109** n o'lchamli ($n > 1$) massiv berilgan. Massivning eng kichik elementidan oldin va eng katta elementidan keyin 0 qiymati joylashtirilsin(1-uchragan).
- 4.110** n o'lchamli massiv hamda 2 ta butun k va m ($0 < k < n$, $0 \leq m < k$) sonlari berilgan. Massivning k -indeksli elementidan oldingi m ta elementiga 0 qiymati o'rnatilsin.
- 4.111** n o'lchamli ($n > 1$) massiv berilgan. Shatrli operatoridan

- foydalanmasdan uning $a_0, a_2, a_4, \dots, a_{n-2}$ elementlari ikkilantirilsin.
- 4.112** n o'lchamli ($n > 0$) massiv berilgan. Shatrli operatoridan foydalanmasdan uning $a_1, a_3, \dots, a_{n-1}$ elementlari uchlantirilsin.
- 4.113** n o'lchamli ($n > 0$) massiv berilgan. Massivning har bir musbat elementi oldidan nol (0) qiymatli element joylashtirilsin.
- 4.114** n o'lchamli ($n > 0$) massiv berilgan. Massivning har bir manfiy elementidan keyin nol (0) qiymatli element joylashtirilsin.
- 4.115** Butun turga tegishli n o'lchamli ($n > 0$) massiv berilgan. Massivdagi juft sonlar ikkilantirilsin.
- 4.116** Butun turga tegishli n o'lchamli ($n > 0$) massiv berilgan. Massivdagi barcha toq sonlar uchlantirilsin.

4.2. Ko'p o'lchovli massivlar

C++ tilida massiv elementlarining turiga cheklovlar qo'yilmaydi, lekin bu turlar chekli o'lchamdagi ob'yektlarning turi bo'lishi kerak. Chunki kompilyator massivning xotiradan qancha joy (bayt) egallashini bilishi zarur. Xususan, massiv komponentasi massiv bo'lishi ham mumkin, ya'ni «vektorlar-vektori» natijada matritsa deb nomlanuvchi ikki o'lchovli massivni hosil qiladi.

Agar matritsaning elementi ham vektor bo'lsa, uch o'lchamli massiv hosil bo'ladi. Shu yo'l bilan ixtiyoriy o'lchamdagi massivni hosil qilish mumkin.

Ikki o'lchovli *statik* massivning sintaksisi quyidagi ko'rinishda bo'ladi:

`<tur><massiv o'zgaruvchisi nomi>[<o'lcham1>][<o'lcham2>];`

Masalan, 5x15 o'lchamli haqiqiy sonlar qabul qiluvchi statik massivning e'loni quyidagicha bo'ladi:

`float s_mas[5][15];`

E'lon qilingan `s_mas` massiv o'zgaruvchisi 5 ta satr va 15 ta ustundan tashkil topgan matritsaga o'xshaydi, ya'ni uning 75 ta elementi mavjud.

Endi adres nuqtai-nazaridan ko'p o'lchovli massiv elementlariga murojaat qilishni ko'raylik. Quyidagi e'lonlar berilgan bo'lsin:

`int a[4][5];`

`float b[3][2][4];`

Birinchi e'londa ikki o'lchovli massiv, ya'ni 4 ta satr va 5 ta ustundan iborat matritsaga o'xshash massiv e'lon qilingan, ikkinchisida

esa uch o'lovli, ya'ni 3 ta satr, 2 ta ustun va 4 ta qavatdan iborat massiv e'lon qilingan.

Massiv elementlariga murojaat qilish uchun massiv nomidan keyin kvadrat qavsda har bir o'lcham uchun indeks yozilishi kerak, *masalan*, $a[i][j]$ ko'rinishidagi asosiy murojaatdan tashqari, vositali murojaat qilish $*(a+i)+j$ yoki $*(a[i]+j)$ va $b[i][j][k]$ ko'rinishidagi asosiy murojaatdan tashqari, vositali murojaat qilish $*(b+i)+j+k$ yoki $*(b[i]+j)+k$ yoki $*(b[i][j]+k)$ ham mumkin.

Ko'p o'lovli massivlarni initsializatsiyalash quyidagi misollar yordamida ko'rsatiladi:

```
int s_mas[3][4]={0,4,1,-5,2,-22,15,-46,18,98,12,-3};
```

```
int b[4][3]={5,1,24,1,34,-12,76,-2,7,7,-25,7};
```

Birinchi e'londa boshlang'ich qiymatlar ketma-ket yozilgan, ikkinchi e'londa esa qiymatlar guruhlashgan.

Endi ikki o'lovli dinamik massivlar tashkil qilishni qarab chiqamiz. Bunday massivlarni tashkil etishning sintaksisi quyidagi:

```
<tur> **<massiv o'zgaruvchisi nomi>;
```

ko'rinishda bo'lgan «ko'rsatkichga ko'rsatkich» ishlatiladi.

Dastlab massiv satrlari soniga qarab ko'rsatkichlar massivga dinamik xotiradan joy ajratiladi:

```
d_mas=new int *[n]; // bu yerda n massiv satrlari soni
```

Keyin, dinamik massivning har bir satri uchun takrorlash operatoridan foydalanib xotiradan joy ajratish va ularning boshlang'ich adreslarini d_mas massiv elementlariga joylashtirish zarur bo'ladi:

```
for(int i=0;i<n;i++) d_mas[i]=new int[m]; //m ustunlar soni
```

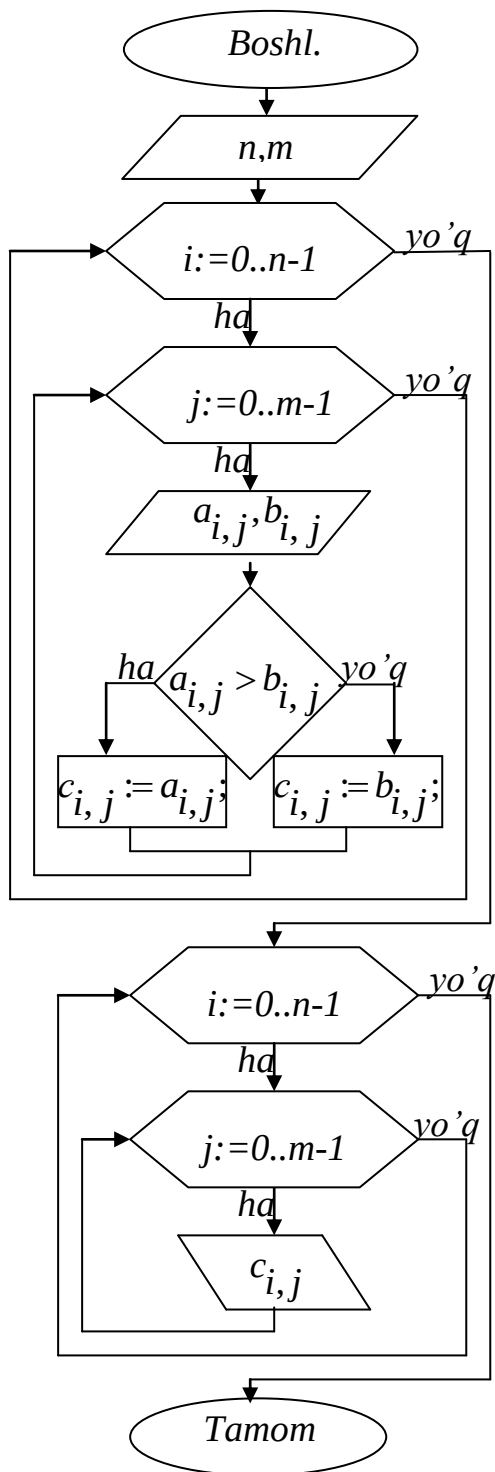
Shuni ta'kidlash kerakki, dinamik massivning har bir satri xotiraning turli joylarida joylashishi mumkin.

Ikki o'lovli dinamik massivni o'chirish uchun esa oldin massivning har bir elementi (satri), so'ngra massivning o'zi o'chiriladi:

```
for (i=0;i<n;i++) delete [] d_mas[i];
```

```
delete [] d_mas;
```

Masala. O'lchamlari $n \times m$ bo'lgan ($n > 0, m > 0$) ikkita A va B matritsalar berilgan, ulardan foydalanib C matritsa tuzilsin: $C_{i,j} = \max(A_{i,j}, B_{i,j})$ (C matritsaning har bir elementi A va B matritsalaridagi mos elementlarning kattasi hisoblanadi). Qo'yilgan masala uchun algoritm va dastur tuzilsin.



```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char
*argv[])
{
    int n,m;
    float a[100][100],b[100][100],
    c[100][100];
    cin>>n>>m;
    for(int i=0;i<=n-1;i++)
    for(int j=0;j<=m-1;j++)
    {
        cin>>a[i][j]>>b[i][j];
        if(a[i][j]>b[i][j])
            c[i][j]=a[i][j];
        else
            c[i][j]=b[i][j];
    }
    for(int i=0;i<=n-1;i++)
    for(int j=0;j<=m-1;j++)
        cout<<c[i][j];

    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Quyidagi masalalar uchun algoritmlar va dasturlar tuzilsin:

A toifa

- 4.117** n va m butun musbat sonlar berilgan. Massivning i -satridagi har bir elementiga $10*i(i=1, \dots, m)$ qiymatlarni joylashtirish bilan $n \times m$ o'lchamli butun sonli matritsa tashkil etilsin.
- 4.118** n va m butun musbat sonlar berilgan. Massivning j -ustunidagi

- har bir elementiga $5*j(j=1, \dots, n)$ qiymatlarni joylashtirish bilan $n \times m$ o'lchamli butun sonli matritsa tashkil etilsin.
- 4.119** n, m butun musbat sonlar va n ta sonlar ketma ketligi berilgan. Har bir ustunida tanlangan sonlar ketma ketligini joylashtirish orqali $n \times m$ o'lchamli matritsa tashkil etilsin.
- 4.120** n, m butun musbat sonlar va m ta sonlar ketma ketligi berilgan. Har bir satrida tanlangan sonlar ketma ketligini joylashtirish orqali $n \times m$ o'lchamli matritsa tashkil etilsin.
- 4.121** n, m butun musbat sonlari, d soni va n ta sondan iborat ketma ketlik berilgan. $n \times m$ o'lchamli matritsa quyidagicha tashkil etilsin: 1-ustunida berilgan sonlar ketma ketligi joylashgan, qolgan ustunlar elementlari o'zidan oldingi ustundagi mos elementlarga d sonini qo'shish orqali hosil qilingan.
- 4.122** n, m butun musbat sonlari, q soni va m ta sondan iborat ketma ketlik berilgan. $n \times m$ o'lchamli matritsa quyidagicha tashkil etilsin: 1-satrida berilgan sonlar ketma ketligi joylashgan, qolgan satrlar elementlari o'zidan oldingi satrdagi mos elementlarga q sonini ko'paytirish orqali hosil qilingan.
- 4.123** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq n-1)$ soni berilgan. Berilgan matritsaning k -satridagi elementlar chop etilsin.
- 4.124** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq m-1)$ soni berilgan. Berilgan matritsaning k -ustunidagi elementlar chop etilsin.
- 4.125** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Juft o'rinda joylashgan satrlardagi elementlar chop etilsin. Satr bo'yicha elementlarini chiqarishda shart operatoridan foydalanilmasin.
- 4.126** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Toq o'rinda joylashgan ustunlardagi elementlar chop etilsin. Ustun bo'yicha elementlarini chiqarishda shart operatoridan foydalanilmasin.
- 4.127** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Uning elementlari quyidagi tartibda chop etilsin: 1-satr elementlarini chapdan o'ngga, 2-satr elementlarini o'ngdan chapga, 3-satr elementlarini chapdan o'ngga, 4-satr elementlarini o'ngdan chapga va hokazo.
- 4.128** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Uning elementlari quyidagi tartibda chop etilsin: 1-ustun elementlarini tepadan pastga, 2-ustun elementlarini pastdan tepaga va hokazo.
- 4.129** $m \times m$ o'lchamli ($m > 0$) A kvadrat matritsa berilgan. Uning

elementlari quyidagi ko‘rinishda chiqarilsin: 1-ustundagi barcha elementlar; oxirgi satrdagi chop etilmagan elementlar; 2-ustundagi qolgan elementlar, oxiridan oldingi satrdagi qolgan element va hokazo. Eng oxirida oxirgi ustunning birinchi elementi chiqadi.

- 4.130** m – tartibli ($m > 0$) A kvadrat matritsa berilgan (m -toq son). Matritsa elementlari soat strelkasi bo‘yicha spiralsimon ko‘rinishda chop etilsin : 1-satr elementlari chapdan o‘ngga qarab, oxirgi ustunning qolgan elementlari yuqoridan pastga qarab, oxirgi satrning qolgan elementlari o‘ngdan chapga qarab, 1-ustunning qolgan elementlari pastdan yuqoriga qarab, 2-satrning qolgan elementlari chapdan o‘ngga qarab va hokazo. Oxirida markazdagi element chop etiladi.
- 4.131** m – tartibli ($m > 0$) a kvadrat matritsa berilgan (m -toq son). Matritsa elementlari soat strelkasiga teskari tartibda spiralsimon ko‘rinishda chop etilsin: 1-ustun elementlari yuqoridan pastga qarab, oxirgi satrning qolgan elementlari chapdan o‘ngga qarab, oxirgi ustunning qolgan elementlari pastdan yuqoriga qarab, 1-satrning qolgan elementlari o‘ngdan chapga qarab, 2-ustunning qolgan elementlari yuqoridan pastga qarab va hokazo. Eng oxirida markazdagi element chop etiladi.
- 4.132** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq n-1)$ butun son berilgan. Berilgan matritsaning k -satridagi elementlarining yig‘indisi va ko‘paytmasi chop etilsin.
- 4.133** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq m-1)$ butun son berilgan. Berilgan matritsaning k -ustunidagi elementlarining yig‘indisi va ko‘paytmasi chop etilsin.
- 4.134** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Uning har bir satri uchun yig‘indilar hisoblansin.
- 4.135** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Uning har bir ustuni uchun yig‘indilar hisoblansin.
- 4.136** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning toq o‘rindagi har bir satrning o‘rta arifmetiklari hisoblansin. (Shart operatoridan foydalanilmasin.)
- 4.137** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir juft o‘rindagi ustuni uchun yig‘indilar hisoblansin. (Shart operatoridan foydalanilmasin.)
- 4.138** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir satridagi eng kichik elementlar topilsin.

- 4.139** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir ustunidagi eng katta elementlar topilsin.
- 4.140** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir satridagi elementlar yig'indisi hisoblanib, yig'indisi eng katta bo'lgan satr nomeri va shu yig'indi chop etilsin.
- 4.141** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir ustunidagi elementlar ko'paytmasi hisoblanib, ko'paytmasi eng kichik bo'lgan ustun nomeri hamda shu ko'paytma chop etilsin.
- 4.142** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir ustuni uchun topilgan eng katta elementlar orasidan eng kichigi topilsin.
- 4.143** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir satri uchun topilgan eng kichik elementlar orasidan eng kattasi topilsin.
- 4.144** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir satri uchun o'rta arifmetik qiymatlar hisoblansin va shu satrdagi o'rta arifmetik qiymatdan kichik bo'lgan elementlarning soni (har bir satr uchun alohida) chiqarilsin.
- 4.145** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir ustuni uchun topilgan o'rta arifmetik qiymatlardan mos ustundagi katta elementlarning soni (har bir ustun uchun alohida) topilsin.
- 4.146** $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning barcha elementlarining o'rta arifmetigiga yaqin qiymatga ega bo'lgan element joylashgan ustun va satr nomerlari chop etilsin. Ularning soni bittadan ko'p bo'lsa birinchisi olinsin.
- 4.147** $n \times m$ o'lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Uning satrlari orasidan musbat va manfiy elementlari soni teng bo'lgan 1-uchragan satr nomeri chop etilsin (0 qiymatli element hisobga olinmasin). Agar bunday satr yo'q bo'lsa, 0 chiqarilsin.
- 4.148** $n \times m$ o'lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Uning ustunlari orasidan musbat va manfiy elementlari soni teng bo'lgan oxirgi uchragan ustun nomeri chop etilsin (0 qiymatli element hisobga olinmasin). Agar bunday ustun yo'q bo'lsa, 0 chiqarilsin.
- 4.149** $n \times m$ o'lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Matritsa satrlari orasidan faqat juft sonlarni o'zida saqllovchi eng oxirgi uchragan satr nomeri chop etilsin. Agar bunday satr

yo‘q bo‘lsa, 0 chiqarilsin.

- 4.150** $n \times m$ o‘lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Matritsaning ustunlari orasidan faqat toq sonlar joylashgan 1-uchragan ustunining nomeri chop etilsin. Agar bunday ustun yo‘q bo‘lsa, 0 chiqarilsin.
- 4.151** $n \times m$ o‘lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Matritsaning elementlarida 0 dan 100 gacha oraliqdagi butun sonlar joylashishi mumkin. Birinchi satrga o‘xshash bo‘lgan satrlar soni topilsin (*Bir satrdagi barcha elementlar 2- satrda ham to‘liq topilsa, bu satrlar “o‘xshash” deyiladi.*).
- 4.152** O‘zida 0 dan 100 gacha oraliqdagi butun sonlarni saqlaydigan $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning oxirgi ustuniga o‘xshash bo‘lgan ustunlar soni topilsin (*Bir ustundagi barcha elementlar 2- ustunda ham to‘liq topilsa, bu ustunlar “o‘xshash” deyiladi.*).
- 4.153** $n \times m$ o‘lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. O‘zida bir xil elementlarni saqlamaydigan satrlari soni topilsin.
- 4.154** $n \times m$ o‘lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. O‘zida bir xil elementlarni saqlamaydigan ustunlari soni topilsin.
- 4.155** $n \times m$ o‘lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Uning ustunlari orasidan eng ko‘p bir xil elementni saqlaydigan 1-uchragan ustun nomeri topilsin.
- 4.156** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Elementlari o‘lish tartibida joylashgan satrlari soni topilsin.
- 4.157** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa berilgan. Elementlari kamayish tartibida joylashgan ustunlari soni topilsin.
- 4.158** $n \times m$ o‘lchamli ($n > 0, m > 0$) butun sonli matritsa berilgan. Matritsada o‘z satrida eng kichik va o‘z ustunida eng katta bo‘lgan element topilsin. Agar bunday element yo‘q bo‘lsa, 0 chiqarilsin.

B toifa

- 4.159** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa va butun k_1 va k_2 ($0 \leq k_1 < k_2 \leq n-1$) sonlar berilgan. Matritsaning k_1 va k_2 satrlari o‘rni almashtirilsin.
- 4.160** $n \times m$ o‘lchamli ($n > 0, m > 0$) matritsa va butun k_1 va k_2 ($0 \leq k_1 < k_2 \leq m-1$) sonlar berilgan. Matritsaning k_1 va k_2 ustunlari o‘rni almashtirilsin.

- 4.161 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning har bir satridagi eng katta va eng kichik elementlarining o'rnini almashtirilsin.
- 4.162 $n \times m$ o'lchamli ($n > 1, m > 1$) matritsa berilgan. Matritsani har bir ustunidagi eng katta va eng kichik elementlarining o'rnini almashtirilsin.
- 4.163 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning eng katta va eng kichik elementlari joylashgan satrlarining o'rnini almashtirilsin.
- 4.164 $n \times m$ o'lchamli ($n > 1, m > 1$) matritsa berilgan. Matritsaning eng katta va eng kichik elementlari joylashgan ustunlarining o'rnini almashtirilsin.
- 4.165 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning 1-ustuni, elementlari faqat musbat sonlardan iborat oxirgi uchragan ustuni bilan almashtirilsin.
- 4.166 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan (n -juft son). Matritsaning yuqori va quyi bo'laklarining o'rinlari almashtirilsin.
- 4.167 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan (m -juft son). Matritsaning o'ng va chap bo'laklarining o'rinlari almashtirilsin.
- 4.168 $n \times m$ ($n > 0, m > 0$) o'lchamli matritsa berilgan (m va n -juft son). Matritsaning yuqori chap va quyi o'ng to'rtidan bir bo'laklari o'rinlari almashtirilsin.
- 4.169 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan (m va n -juft son). Matritsaning quyi chap va yuqori o'ng to'rtidan bir bo'laklarining o'rinlari almashtirilsin.
- 4.170 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan (n -juft son). Matritsa gorizontal simmetrik o'q atrofida 180° burchak ostida burib chop etilsin.
- 4.171 $n \times m$ o'lchamli ($n > 1, m > 1$) matritsa berilgan (m -juft son). Matritsa vertikal simmetrik o'q atrofida 180° burchak ostida burib chop etilsin.
- 4.172 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq n-1)$ butun son berilgan. Matritsaning k -satri o'chirilsin.
- 4.173 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq m-1)$ butun son berilgan. Matritsaning k -ustuni o'chirilsin.
- 4.174 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning eng kichik elementi joylashgan 1-uchragan satr o'chirilsin.

- 4.175 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Matritsaning eng katta elementi joylashgan 1-uchragan ustun o'chirilsin.
- 4.176 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. O'zida faqat musbat sonlarni saqlaydigan 1-uchragan ustun o'chirilsin. Agar bunday ustun topilmasa matritsaning o'zi chop etilsin.
- 4.177 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. O'zida faqat manfiy sonlarni saqlaydigan oxirgi uchragan ustun o'chirilsin. Agar bunday ustun topilmasa matritsaning o'zi chop etilsin.
- 4.178 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq n-1)$ butun son berilgan. matritsaning k -satridan oldin elementlari 0 lardan iborat satr qo'shilsin.
- 4.179 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa va $k(0 \leq k \leq m-1)$ butun son berilgan. Matritsaning k -ustunidan keyin elementlari 1 lardan iborat ustun qo'shilsin.
- 4.180 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Bu matritsaning eng katta elementini saqlovchi 1-uchragan satri ikkilantirilsin.
- 4.181 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Bu matritsaning eng kichik elementini saqlovchi oxirgi uchragan ustuni ikkilantirilsin.
- $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Berilgan matritsaning barcha lokal minimumlari 0 lar bilan almashtirilsin. Masalani yechishda yordamchi matritsadan foydalanishga ruxsat etiladi (*Matritsaning elementi o'zini o'rab turuvchi elementlar orasida eng kichigi bo'lsa, u matritsaning lokal minimumi deyiladi*).
- 4.182 $n \times m$ o'lchamli ($n > 0, m > 0$) matritsa berilgan. Berilgan matritsaning barcha lokal maksimumlari 0 lar bilan almashtirilsin. Masalani yechishda yordamchi matritsadan foydalanishga ruxsat etiladi (*Matritsaning elementi o'zini o'rab turuvchi elementlar orasida eng kattasi bo'lsa, u matritsaning lokal maksimumi deyiladi*).
- 4.183 m -tartibli ($m > 0$) kvadrat matritsa berilgan. Uning bosh dioganali elementlari yig'indisi hisoblansin.
- 4.184 m -tartibli ($m > 0$) kvadrat matritsa berilgan. Matritsaning yordamchi dioganali elementlari o'rta arifmetigi hisoblansin.
- 4.185 m -tartibli ($m > 0$) kvadrat matritsa berilgan. Uning bosh dioganaliga parallel bo'lgan har bir diagonal uchun yig'indilar hisoblansin.
- 4.186 m -tartibli ($m > 0$) kvadrat matritsa berilgan. Uning yordamchi

diagonaliga parallel bo'lgan har bir diagonal uchun yig'indilar hisoblansin.

- 4.187** m -tartibli ($m > 0$) kvadrat matritsa berilgan. Uning bosh diagonalining quyi qismida yotuvchi elementlari nol (0)lar bilan almashtirilsin. Shart operatoridan foydalanilmasin.
- 4.188** m -tartibli ($m > 0$) kvadrat matritsa berilgan. Uning yordamchi diagonalining yuqori qismida yotuvchi elementlari nol (0)lar bilan almashtirilsin. Shart operatoridan foydalanilmasin.
- 4.189** m -tartibli ($m > 0$) kvadrat matritsa berilgan. Matritsaning yordamchi diagonal va undan pastda joylashgan barcha elementlari nol (0)lar bilan almashtirilsin. Shart operatoridan foydalanilmasin.
- 4.190** m -tartibli ($m > 0$) kvadrat matritsa berilgan. Matritsaning bosh diagonal va undan yuqorida joylashgan barcha elementlari nol (0)lar bilan almashtirilsin. Shart operatoridan foydalanilmasin.
- 4.191** m -tartibli ($m > 2$) kvadrat matritsa berilgan. Matritsaning bir vaqtda bosh va qo'shimcha diagonalalaridan yuqorida joylashgan elementlari nol (0)larga tenglashtirilsin. Shart operatoridan foydalanilmasin.
- 4.192** m -tartibli ($m > 2$) kvadrat matritsa berilgan. Matritsaning bir vaqtda bosh diagonal dan yuqori; hamda yordamchi diagonal dan quyi qismida joylashgan elementlari nol (0)lar bilan almashtirilsin.
- 4.193** m -tartibli ($m > 2$) kvadrat matritsa berilgan. Matritsaning bir vaqtda bosh diagonal dan quyi(o'zi ham) va yordamchi diagonal dan yuqori(o'zi ham) qismida yotuvchi elementlari nol (0)larga tenglashtirilsin. Shart operatoridan foydalanilmasin.
- 4.194** m -tartibli ($m > 2$) kvadrat matritsa berilgan. Matritsaning bir vaqtda bosh va yordamchi diagonalalaridan quyi qismida joylashgan elementlari nol (0)lar bilan almashtirilsin.
- 4.195** m -tartibli ($m > 0$) a kvadrat matritsa berilgan. Matritsa bosh diagonal atrofida 180^0 ga burilsin. Yordamchi matritsadan foydalanilmasin.
- 4.196** m -tartibli ($m > 0$) a kvadrat matritsa berilgan. Matritsa yordamchi diagonal atrofida 180^0 ga burilsin. Yordamchi matritsadan foydalanilmasin.
- 4.197** m -tartibli ($m > 0$) a kvadrat matritsa berilgan. Matritsa soat strelkasi bo'yicha 180^0 ga burilsin. Yordamchi matritsadan foydalanilmasin.

- 4.198** m -tartibli ($m > 0$) a kvadrat matritsa berilgan. Matritsa soat strelkasiga teskari yo'nalish bo'yicha 90^0 ga burilsin. Yordamchi matritsadan foydalanilmasin.

4.3. Bobga doir qiyinroq masalalar

- 4.199** *Tarif [11].*

Bosh shifokor uzoq davom etgan jarrohlik amaliyotidan so'ng xodimlarni uyiga olib borish uchun taksi buyurtma berishga qaror qildi. U xodimlar soniga teng bo'lgan N ta taksiga buyurtma berdi. Har bir taksining 1 kilometr masofa uchun o'z tarifi borligi ma'lum. Bosh shifokor har bir xodimni ishdan uyigacha bo'lgan masofani biladi (lekin har bir xodim har xil yo'nalishli joylarda yashashadi, ya'ni ikkita xodimni bitta taksida jo'natishni imkoni yo'q). Endi Bosh shifokor qaysi xodimni qaysi taksida uyiga jo'natish masalasini hal qilishi kerak. Ularni shunday jo'natish kerakki, jami taksiga to'langan summa eng minimal bo'lsin.

Berilganlar:

Shifoxanadagi xodimlar sonini bildiruvchi N ($1 \leq N \leq 1000$) soni berilgan (o'z navbatida taksilar soni ham shuncha). Keyingi satrda xodimlarni shifoxanadan uyigacha bo'lgan masofani bildiruvchi N ta son berilgan (birinchi son – birinchi xodimniki, ikkinchi - ikkinchi xodimniki va h.). Keyingi satrda taksilarning tarifini bildiruvchi N ta son berilgan (birinchi son – birinchi taksiniki, ikkinchi - ikkinchi taksiniki va h.). Ta'riflar butun son bo'lib, 10000 dan oshmaydi.

Natija:

N ta son chiqarilsin. Birinchi son - birinchi xodim o'tirgan taksi nomeri, Ikkinchi son–ikkinchi xodim o'tirgan taksi nomeri va hokazo. Taksilarga sarflangan minimal summa.

<i>Berilganlar:</i>	<i>Natija:</i>
3 10 20 30	1 3 2

50 20 30	
5	5 1 3 2 4
10 20 1 30 30	
3 3 3 2 3	

4.200

Natural sonlar

Ko'pi bilan 24 ta raqamdan tashkil topgan N natural son berilgan. N sonining birinchi soni 0 dan farqli. Raqamlarni o'rnini almashtirgan holda N dan kichik bo'lgan nechta son hosil qilish mumkin.

Masalan: 2007(0ta), 2515(4 ta 2155,1552,1525,1255)

Berilganlar:

N soni berilgan.

Natija:

Raqamlarni almashtirish natijasida hosil bo'lgan N dan kichik bo'lgan sonlar miqdori chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
2007	0
2515	4

4.201

Poyezd vagonlari [11].

Vali saralash algoritmlarini o'rganishga qiziqadi. U bir kuni o'sish tartibida nomerlangan poyezd vagonlarini ko'rib qoldi va shu vagonlarni kamayish tartibida joylashtirish uchun qanday harakatlar bajarish kerakligini o'yladi. Harakatning k -bosqichida vagonlar tartibi qanday bo'lishi bilan ham qiziqdi. Valiga k -saralash bosqichidagi vagonlar tartibini aniqlashga yordam bering.

Berilganlar:

Ikkita n va k sonlari berilgan ($1 < n < 30$, $1 < k < 10^{18}$).

Natija:

N ta butun son chiqarilsin: saralalashning k -bosqichidagi vagonlar tartibi. Agarda saralashlar soni k dan kichik bo'lsa -1 chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
---------------------	----------------

4 1	12 3 4
4 3	13 2 4
4 15	-1

4.202

Turli natijalar.

Bir kuni xonasini tartibga solayotgan Vali boshlang'ich maktabda yozgan kundalik daftarini topib oldi. Undagi $2+2=2*2$ struktura uni ajablantirdi. Nahotki, ko'paytirish va qo'shish bir xil natija bersa-ya!

Bu struktura Valini quyidagi masalaning tug'ilishiga turtki bo'ldi: ikkita a va b sonlari berilgan bo'lsa, quyidagi sonlar to'plami nechta har xil qiymat beradi

$$a+b, a-b, a*b, a/b, a^b, b+a, b-a, b*a, b/a, b^a$$

Bo'lish yaxlitlashsiz amalga oshiriladi. Bo'lish natijasi butun son bo'lmasligi ham mumkin. Agarda to'plamdagi qaysidir ifoda noto'g'ri bo'lsa, Vali uni qarab o'tirmaydi. Nolga bo'lish va nolni manfiy bo'lmagan darajaga ko'tarish noto'g'ri ifoda hisoblanadi

Berilganlar:

Ikkita a va b ($|a|, |b| \leq 10^9$) sonlari berilgan.

Natija:

Berilgan to'plam boyicha har xil qiymatlar soni chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
2 5	8
0 1	3

4.203

Sonlar darchasi [11].

Butun sonlardan iborat uzunligi N ga teng ketma-ketlik bo'ylab 1 qadam bilan K uzunlikdagi "darcha" harakatlanmoqda, ya'ni boshlanishda "darcha" da dastlabki K ta son ko'rinadi, keyingi qadamda "darcha" da ikkinchi sondan boshlab K ta ko'rinadi va bu holat oxirgacha davom etadi. "Darcha" ning har bir holati uchun undagi eng kichik qiymat topilsin.

Berilganlar:

Birinchi satrda N va K ($1 \leq N \leq 150000$, $1 \leq K \leq 10000$, $K \leq N$) —sonlari berilgan, ular mos ravishda ketma-ketlik va "darcha" uzunligini bildiradi. Keyingi qatorda N ta sondan iborat ketma-ketlik berilgan.

Natija:

Har bir "darcha" uchun eng kichik qiymat alohida satrlarda chiqarilsin.

Berilganlar:	Natija:
7 3	1
1 3 2 4 5 3 1	2
	2
	3
	1

4.204

O'nli kasr [11].

Natural son berilgan. $1/n$ kasrning o'nli kasr ko'rinishidagi shakli chiqarilsin.

Berilganlar:

n ($1 \leq n \leq 10\,000$) soni berilgan.

Natija:

$1/n$ kasrning o'nli kasr ko'rinishidagi shakli chiqarilsin.

Berilganlar:	Natija:
1	1
2	0.5
3	0.(3)
14	0.0(714285)

4.205

Kartridj [11].

Vali k sahifali ma'ruzani bosmaga chiqarishi kerak. Afsuski, yotoqxonadagi printerlarning barchasining kartridji ishdan chiqibdi.. Endi Vali yangi kartridj sotib olishi lozim. Magazinda n turdagi kartridjlar bor ekan. Sotuvchi Valiga kartridjlar asosan ikki xil parametrga ega ekanligini tushuntirdi, ya'ni kartridj narxi va bosmaga chiqaradigan varaqlar soni. Ma'lum bo'ldiki, i -chi turdagi kartridj narxi c_i va bosmalaydigan varaqlar soni p_i

ekan. Magazinda har bir turdagi kartridjlardan chegaralanmagan miqdorda mavjud.

Vali oddiy talaba bo'lsa, shuning uchun u shunday arzon kartridjni sotib olishi kerakki, uning ma'ruzasini bosmalashga yetsin. Ikkinchi tomondan esa Vali juda ham xasis. Agarda Vali ma'ruzasini bosmalagandan so'ng atigi bitta varaqni bosmalash mumkin bo'lgan rang kartridjda qolsa, yotoqxonadagilar uning oldiga hujjatlarini bosmalashga olib kelishlarini biladi.

Shuning uchun, Vali narxi eng minimal bo'lgan va k varaqni pechat qilishga teng bo'lgan kartridgni sotib olishni xoxlaydi. Valiga minimal summaga kardtridj olishga va xasislik qilmasligiga yordam bering.

Berilganlar:

Birinchi satrda ikkita n va k sonlari berilgan ($1 < n < 100000$, $1 < k < 109$). Mos ravishda magazindagi kartridj turlari va ma'ruzadagi varaqlar soni. Keyingi n ta satrda esa har bir kartridjning narxi va bosmalashi mumkin bo'lgan maksimal varaqlar soni berilgan. Masalan, i -chi kardtridjniki c_i va p_i ($1 < c_i, p_i < 200$).

Natija:

k varaqni chiqarishi uchun Vali sarflashi mumkin bo'lgan eng minimal summa chiqarilsin. Agarda bunday yechim mavjud bo'lmasa -1 chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
4 5 5 5 2 3 5 10 1 1	4
1 2 1 3	-1

4.206

Kesmadagi nuqtalar.

Sonlar o'qida n ta kesma va kesmalarda m ta nuqta berilgan. Berilgan har bir nuqta nechta kesmada yotishini aniqlang.

Agarda $\min(a,b) < x < \max(a,b)$ shart bajarilsa x nuqta oxiri a va b bo'lgan kesmada yotadi.

Berilganlar:

Birinchi satrda ikkita n ($1 < n < 105$) - kesmalar va m ($1 < m < 105$) nuqtalar soni berilgan.

Keyingi n ta satrda kesmalarni boshi oxirining koordinatalari a_i va b_i berilgan. Eng oxirgi satrda m -nuqtaning koordinatasi berilgan. Barcha sonlar moduli boyicha 109 dan oshmaydi.

Natija:

Har bir nuqtaning nechta kesmada yotishini bildiruvchi m ta son chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
2 2	
0 5	
7 10	
1 6	

4.207

Do'stona juftliklar[12]

Yo'ldosh, Shavkat va G'iyos topib olingan massivdagi do'stona juftliklarni aniqlashga qaror qilishdi. a massiv indeksleri 1 dan n gacha nomerlangan. $a[i]$ va $a[j]$ juftliklar do'stona bo'lish uchun Yo'ldosh $a[i] \% a[j] = 0$, G'iyos $i \% j = 0$, Shavkat esa $i > j$ bo'lishi kerak dedi. Ularga uchalasining shartini qanoatlantiradigan juftliklar sonini topishda yordam bering.

Berilganlar:

Birinchi qatorda n massiv elementlari soni berilgan ($1 \leq n \leq 105$). Ikkinchi qatorda n ta butun son massiv elementlari bitta bo'sh joy bilan ajratilgan holda berilgan. Massiv elementlari qiymatlari 1 dan $2 \cdot 10^9$ gacha bo'ladi.

Natija:

Do'stona juftliklar sonini chiqaring.

<i>Nº</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	6 2 3 5 6 10 6	5
2	1 3	0

4.208

Eng kichik uchramaydigan son[12].

Bir o'lchamli sonli massiv berilgan. Sizning vazifangiz unda uchramaydigan eng kichik natural sonni topuvchi dastur

tuzishdan iborat.

Berilganlar:

Birinchi qatorda bitta natural son n –massivdagi elementlar soni ($1 \leq n \leq 105$) berilgan. Ikkinchi qatorda n ta butun son–massiv elementlari bitta bo'sh joy bilan ajratib berilgan. Massiv elementlari qiymatlari -109 dan 109 gacha bo'lishi mumkin.

Natija:

Massivda uchramagan eng kichik natural sonni chiqaring.

<i>Berilganlar:</i>	<i>Natja:</i>
5 5 4 1 2 3	6
11 3 -8 -2 1 4 3 1 2 3 8 10	5
1 0	1

4.209

Qayta qiymatlash [12].

Bir o'lchovli butun sonlardan iborat massiv berilgan. Uning elementlarining qiymatlari -109 dan 109 gacha bo'lishi mumkin. Ma'lumki, undagi har xil sonlar miqdori uning elementlar soni n dan oshmaydi. Massiv elementlariga sonlarni qayta qiymatlashda quyidagi shartlar bajarilishi kerak:

- 1) Qayta qiymatlashda ishlatiladigan eng minimal son 1 bo'lishi kerak;
- 2) Qayta qiymatlashda ishlatiladigan maksimal son iloji boricha eng kichik bo'lishi kerak;
- 3) Bir xil qiymatli elementlarning qayta qiymatlangach ham qiymatlari bir xil bo'lishi kerak;
- 4) Avvalgi qiymati katta bo'lgan songa kattaroq qiymat berilishi kerak.

Qayta qiymatlash orqali elementlarning asl qiymati yo'qoladi, lekin ularning tenglik, katta-kichiklik haqidagi ma'lumot saqlanib qoladi. Sizning vazifangiz yuqoridagi shartlarni qanoatlantiradigan tartibda berilgan bir o'lchamli massiv elementlarini qayta qiymatlab chiqishdan iborat. Lekin sizdan massivning barcha elementlarini chiqarish emas, faqat ba'zi elementlarini chiqarish so'raladi.

Berilganlar:

Birinchi qatorda massiv elementlari soni n berilgan ($1 \leq n \leq 105$). Ikkinchi qatorda n ta son–massiv elementlari bitta bo'sh joy

bilan ajratib berilgan. Massiv elementlari butun va modul jihatdan 109 dan oshmaydi. Uchinchi qatorda so'rovlar soni $m(1 \leq m \leq 300)$ beriladi. Keyingi qatorda m ta butun son i -qayta qiymatlangan massivdagi so'ralgan element indeksi beriladi ($1 \leq k_i \leq n$).

Natija:

Bitta qatorda m ta sonni har bir so'rovga javob sifatida qayta qiymatlangan massivdagi i -indeksdagi qiymatini so'rovlar berilish tartibida bitta bo'sh joy bilan ajratib chiqaring.

Berilganlar:	Natija:
10 -8 1 -4 6 1 6 6 9 -4 -8 10 1 2 3 4 5 6 7 8 9 10	1 3 2 4 3 4 4 5 2 1

4.210

Maksimal satr va ustun [12].

$n \times m$ matritsa berilgan. Undan shunday satr va ustunni topingki, bu satrga yoki bu ustunga tegishli bo'lgan matritsa elementlari yig'indisi maksimal bo'lsin. Shu maksimal yig'indini toping.

Berilganlar:

Birinchi qatorda n va m butun sonlari—matritsaning satr va ustunlari soni berilgan ($1 \leq n, m \leq 700$). Keyingi n ta satrning har birida m ta sondan matritsa elementlari bitta bo'sh joy bilan ajratib berilgan. Matritsa elementlari butun va moduli bo'yicha 106 dan oshmaydi.

Natija:

Masalaning javobini chiqaring.

№	Berilganlar:	Natija:
1	3 4 4 -5 3 2 7 8 4 -1 2 3 7 -3	18
2	1 1 2	2

4.211

Bir xil juftliklar [12].

Birinchi kurs talabalari C++ dasturlash tilini o'rganishni boshladi. Ular darsda sikllar va massivlarni o'rganishdi. Dostkaga quyidagicha dastur qismi yozildi:

`long long k = 0;`

```

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        if (a[i]==a[j])
            k++;
    }
}
cout<<k;

```

Ko'pchilik talabalar buning a massivdagi bir xil juftliklar sonini topuvchi dastur ekanligini topishdi. Sizning vazifangiz berilgan a massiv uchun dasturdagi k sonining qiymatini topish.

Berilganlar:

Birinchi qatorda n butun soni– massiv elementlari soni berilgan ($1 \leq n \leq 105$). Ikkinchi qatorda n ta butun son –massiv elementlari bitta bo'sh joy bilan ajratib berilgan. Massiv elementlari modul jihatdan 109 dan oshmaydi.

Natija:

Yagona son – k sonining qiymati chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
6	10
5 -5 2 5 2 1	

4.212

Kasrlarni saralash [12].

Sizga n ta kasr o'zining surat va maxrajining qiymati orqali berilgan. Sizning vazifangiz bu kasrlarni qiymati bo'yicha o'sish tartibida saralashdan iborat. Agar bir nechta kasrning qiymatlari teng bo'lsa ularning bir-biriga nisbatan tartibi berilgan tartibda qoldirilsin.

Berilganlar:

Birinchi qatorda n natural soni berilgan ($1 \leq n \leq 100$). Keyingi n ta qatorda har biridanavbatdagi kasrning surat va mahraji bitta bo'sh joy bilan ajratib berilgan. Surat va mahrajqiymatlari 1 dan 109 gacha bo'lishi mumkin.

Natija:

Dastlabki n ta qatorda saralangan kasrlarning surat va maxrajlarini bitta bo'sh joy bilan ajratib chiqaring.

<i>Berilganlar:</i>	<i>Natija:</i>
4	2 10
1 2	1 3
1 3	1 2
2 4	2 4
2 10	

4.213

Kitoblar [12].

Vali ko'p pul yig'ib, kitob sotib olish maqsadida kitob do'koniga keldi. Unda X so'm pul bor. Do'konda n ta kitob bor. Ularning narxlari massiv ko'rinishida berilgan. U kitobning farqiga bormaydi, shuning uchun uning maqsadi puli yetgancha maksimal sondagi kitob sotib olish. Agar maksimal sondagi kitob olishning bir nechta varianti bo'lsa, u o'zida maksimal pul ortib qoladigan variantni tanlaydi. Unga bu ishda yordam bering.

Berilganlar:

Birinchi qatorda ikkita butun son – n va X sonlari bitta bo'sh joy bilan ajratib yoziladi ($1 \leq n \leq 4 \cdot 10^6$, $1 \leq X \leq 1018$). Ikkinchi qatorda n ta butun sonlar, ya'ni kitoblarning narxlari bittadan bo'sh joylar bilan ajratib yoziladi. Kitoblarning narxlari 1 dan 109 so'mgacha bo'lishi mumkin, lekin ularning maksimali va minimali orasidagi farq 105 dan oshmaydi.

Natija:

Ikkita son – sotib olish mumkin bo'lgan maksimal kitoblar miqdori va talabada qolgan pul miqdori chiqarilsin.

<i>N_o</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	4 6 3 2 1 4	3 0
2	5 12 2 1 5 3 3	4 3
3	2 5 7 7	0 5

V Bob. Belgilar va satrlar

5.1. Belgilar

C++ tilining standartlashgan versiyalarida ikki xil turdagi belgilar majmuasi bilan ishlash imkoniyati mavjud. Birinchi turga, an'anaviy, «tor» belgilar deb nomlanuvchi 8 bitli belgilar majmuasi, ikkinchi turga esa 16 bitli «keng» belgilar majmuasi kiradi. Dasturlash tili kutubxonasida har bir guruh belgilari uchun maxsus funksiyalar to'plami aniqlangan.

Quyida C++ tilida belgi sifatida ishlatilishi mumkin bo'lgan o'zgarmaslar to'plamining jadvali keltirilgan:

<i>Belgi sinflari</i>	<i>Belgi o'zgarmaslar</i>
Katta harflar	'A'..'Z', 'A'..'Я'
Kichik harflar	'a'..'z', 'a'..'я'
Raqlamlar	'0'..'9'
Bo'sh joy	gorizontal tabulyatsiya (ASCII kodi 9), satrni o'tkazish (ASCII kodi 10), vertikal tabulyatsiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12) karetkani qaytarish (ASCII kodi 13).
Punktulasiya belgilari (ajratuvchilar)	! " # \$ % & ' () * + - , . / : ; < = > ? @ [\] ^ _ { } ~
Boshqaruv belgilari	ASCII kodi 0..1Fh oralig'ida va 7Fh bo'lgan belgilar
Bo'sh joy	ASCII da kodi 32 bo'lgan belgi
O'n oltilik raqlamlar	'0'..'9', 'A'..'F', 'a'..'f'

5.1-jadval. C++ tilida belgi sifatida ishlatilishi mumkin bo'lgan o'zgarmaslar to'plami.

Belgi va satrlarni o'qish-yozish funksiyalari makros ko'rinishida aniqlangan.

`getc()` makrosi tayinlangan oqimdan navbatdagi belgini qaytaradi va kirish oqimi ko'rsatkichini keyingi belgiga o'tkazadi. Agar o'qish muvaffaqiyatli bo'lsa `getc()` funksiyasi ishorasiz `int` ko'rinishidagi qiymatni, aks holda `EOF` qiymatini qaytaradi. Ushbu funksiya prototipi quyidagicha:

```
int getc(FILE * stream);
EOF identifikator makrosi
#define EOF(-1);
```

ko'rinishida aniqlangan va o'qish-yozish amallarida fayl oxirini belgilash uchun xizmat qiladi. *EOF* qiymati ishorali *char* turida bo'ladi. Shu sababli o'qish-yozish jarayonida *unsigned char* turidagi belgilar ishlatilsa, *EOF* makrosini ishlatib bo'lmaydi.

Belgini oqimga chiqarish uchun *putc()* makrosi aniqlangan va uning prototipi quyidagi:

```
int putc(int c, FILE*stream);
```

ko'rinishida aniqlangan. *putc()* funksiyasi *stream* nomi bilan berilgan oqimga *c* belgini chiqaradi. Funksiya qaytaruvchi qiymat *int* turiga aylantirilgan *c* belgi hisoblanadi. Agar belgini chiqarishda xatolik ro'y bersa *EOF* qaytariladi.

getc() va *putc()* makroslarining qo'llanilishiga doir misol:

```
#include <iostream.h>
#include <stdio.h>
int main()
{
    char ch;
    cout<<"Belgini kiriting:";
    ch=getc(stdin);
    cout<<"Siz kiritgan belgi:\n";
    putc(ch,stdout);
    return 0;
}
```

Belgilarning *ASCII* jadvalidagi kodini chop etish yoki berilgan kodga mos jadvaldagi belgini chiqarish uchun turni oshkor ko'rinishda boshqa turga keltirishdan foydalaniladi.

Masala. *ASCII* jadvalidagi barcha belgilarni o'z kodlari bilan birga chop etadigan dastur tuzilsin.

```
#include <iostream.h>
int main()
{
    for (int i=0;i<=255;i++)
        Cout<<i<<"|t"<<(char)i<<"|n";
    Return 0;
}
```

Dastur bajarilganda *ASCII* jadvalidagi barcha kodlar va ularga mos belgilar chop etiladi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 5.001 Ixtiyoriy belgi berilgan. Uning ASCII jadvalidagi kodi chop etilsin.
- 5.002 $n(32 \leq n \leq 126)$ butun soni berilgan. ASCII jadvalidagi kodi n ga teng bo'lgan belgi chop etilsin.
- 5.003 Ixtiyoriy belgi berilgan. Bu belgidan (jadvalda) oldin va keyin joylashgan 2 ta belgi chop etilsin.
- 5.004 $n(1 \leq n \leq 26)$ soni berilgan. Lotin alfavitidagi birinchi n ta bosh hartflar chop etilsin.
- 5.005 $n(1 \leq n \leq 26)$ soni berilgan. Lotin alfavitidagi oxirgi n ta kichik hartflar chop etilsin.
- 5.006 Ixtiyoriy belgi berilgan. Agar u raqam bo'lsa, "*raqam*", lotin harfi bo'lsa "*lotin*" aks holda "*belgi*" xabari chop etilsin.
- 5.007 Bo'sh bo'lmagan satr berilgan. Uning birinchi va oxirgi belgilarining ASCII jadvalidagi kodlari chop etilsin.
- 5.008 $n(n > 0)$ butun son va ixtiyoriy belgi berilgan. Bu belgidan tuzilgan n uzunlikka ega satr chiqarilsin.
- 5.009 $n(n > 1)$ juft soni va 2 ta ixtiyoriy belgi berilgan. Birinchi belgidan boshlanib navbat bilan almashib keluvchi n uzunlikka ega, shu belgilardan tuzilgan ketma-ketlik chop etilsin.

5.2. Satrlar

C++ tilida satr *char* turidagi belgilar massivi sifatida qaraladi va bu belgilar ketma-ketligi *satr terminatori* deb nomlanuvchi 0 kodli belgi bilan tugaydi (`\0`). *Nol-terminator* bilan tugaydigan satrlarni *ASCIIz-satlar* deyiladi.

Satr massivi e'lon qilinishida, satr oxiriga terminator qo'yilishi va natijada satrga qo'shimcha bir bayt qo'shilishini hisobga olish kerak:

```
char satr[10];
```

Ushbu e'londa satr o'zgaruvchisiga xotiradan 10 bayt joy ajratiladi, bundan 9 bayti satr hosil qiluvchi belgilar uchun va 1 bayti esa nol-terminatori uchun.

Satr o'zgaruvchilar e'lon qilinishida boshlang'ich qiymatlarni qabul qilishi mumkin. Bu holda kompilyator avtomatik ravishda satr uzunligini hisoblaydi va satr oxiriga nol-terminatorini qo'shib qo'yadi:

```
char Hafta_kuni[] = "Juma";
```


Ushbu e'lon quyidagi e'lon bilan ekvivalent:

```
char Hafta_kuni[] = { 'J', 'u', 'm', 'a', '\0' } ;
```

Satr qiymatini o'qishda oqimli o'qish operatori ">>" o'rniga *getline()* funksiyasini ishlatish ma'qul hisoblanadi, chunki oqimli o'qishda bo'sh joylar inkor qilinadi (ular satr belgisi hisoblansa ham). Oqimli o'qishda o'qilayotgan belgilar ketma-ketligi satrdan o'lchamidan oshib ketganda ham belgilarning kiritilishi davom etishi mumkin. Natijada satr o'ziga ajratilgan o'lchamdan ortiq belgilarni qabul qiladi va turli xatoliklar kelib chiqadi. Shu sababli ham satrlarni o'qishda *getline()* funksiyasidan foydalanish maqsadga muvofiq. *getline()* funksiyasi ikkita parametrga ega bo'lib, birinchi parametr o'qish amalga oshirilayotgan satrga ko'rsatkich, ikkinchi parametr esa o'qilishi kerak bo'lgan belgilar sonini ko'rsatadi.

Oqimdan satrni o'qishga mo'ljallangan *gets()* funksiyasining prototipi quyidagi,

```
char * gets(char *s);
```

ko'rinishida aniqlangan bo'lib, u standart oqimdan satrni o'qiydi va uni *s* o'zgaruvchisiga yozadi. Joylashtirish paytida oqimdagi '\n' belgisi '\0' belgisi bilan almashtiriladi. Bu funktsiyani ishlatishda o'qilayotgan satrning uzunligi *s* satr uchun ajratilgan joy uzunligidan oshib ketmasligini nazorat qilish kerak bo'ladi.

puts() funksiyasi quyidagi

```
int puts(const char *s);
```

ko'rinishda bo'lib, u standart oqimga argumentda ko'rsatilgan satrni chiqaradi. Bunda satr oxiriga yangi satrga o'tish belgisi '\n' qo'shiladi. Agar satrni oqimga chiqarish muvaffaqiyatli bo'lsa *puts()* funksiyasi manfiy bo'lmagan sonni, aks holda *EOF* ni qaytaradi.

Satrni o'qish-yozish funksiyalarini ishlatishga misol tariqasida quyidagi dasturni keltirish mumkin.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char *s;
```

```
    puts("Satrni kiriting: ");
```

```
    gets(s);
```

```
    puts("Kiritilgan satr: ");
```

```
    puts(s);
```

```
    return 0;
```

```
}
```

Satr bilan ishlaydigan funksiyalarning aksariyati «*string.h*» kutubxonasida jamlangan. Dasturlashda nisbatan ko'p ishlatiladigan funksiyalarning tavsifini keltiramiz:

strlen()—satr uzunligini aniqlash funksiyasi.

Sintaksisi: *size_t strlen(const char* string);*

Bu funksiya uzunligi hisoblanayotgan satr boshiga ko'rsatkich bo'lgan yagona parametrga ega va u natija sifatida ishorasiz butun sonni qaytaradi. *strlen()* funksiyasi satrning haqiqiy uzunligidan bitta kam qiymat qaytaradi, ya'ni u *nol-terminatorning* o'rnini hisobga olmaydi. Satr uzunligini aniqlash uchun *sizeof()* funksiyasidan ham foydalanish mumkin va u *strlen()* funksiyasidan farqli ravishda satrning haqiqiy uzunligini qaytaradi.

sizeof() funksiyasidan *getline()* funksiyasining ikkinchi argumenti sifati foydalaniladi va u satr uzunligini yaqqol ko'rsatmaslik imkonini beradi:

cin.getline(Satr, sizeof(Satr));

Satr o'zgaruvchilarining qiymatlarini biridan ikkinchisiga nusxalash mumkin. Bu maqsadda bir qator standart funksiyalar aniqlangan bo'lib, ularning ayrimlari bilan tanishamiz.

strcpy()—satrni nusxalash funksiyasi.

Sintaksisi: *char* strcpy(char* str1, const char* str2);*

Bu funksiya *str2* satrdagi belgilarni *str1* satriga baytma-bayt nusxalaydi.

Satrlarni ulash (konkatenatsiya) funksiyalari *strcat()* va *strncat()*.

Sintaksislari: *char* strcat(char* str1, const char* str2);* Bu funksiya *str1* satri davomidan *str2* satrini ulaydi.

char strncat(char* str1, const char* str2, size_t num);* Bu funksiya esa *str1* satri davomidan *str2* satrining 'num'ta belgisini ulaydi.

Satrdagi harf registrilarini o'zgartirish funksiyalari *_strupr()* va *_strlwr()*. Kompilyatorlarning ayrim variantlarida funksiyalar nomidagi tagchiziq ('_') bo'lmasligi ham mumkin.

Sintaksisi: *char _strlwr(char* str);* argument sifatida berilgan satrdagi bosh harflarni kichik harflarga almashtiradi va hosil bo'lgan satr adresini funksiya natijasi sifatida qaytaradi.

_strupr() funksiyasi xuddi *_strlwr()* funksiyasidek amal qiladi, lekin satrdagi kichik harflarni bosh harflarga almashtiradi:

Satrni teskari tartiblash funksiyasi *strrev()*.

Sintaksisi: *char* strrev(char* str);* str satridagi belgilarni teskari tartibda tartiblaydi.

Turlarni o'zgartirish funksiyalari: `atoi()`—belgi ko'rinishidagi butun sonni butun son ko'rinishiga o'tkazish funksiyasi.

Sintaksisi: `int atoi(const char* ptr);` `ptr` ko'rsatgichdagi ASCIIz-satrn int turidagi songa o'tkazishni amalga oshiradi.

`atol()` funksiyasi xuddi `atoi()` funksiyasidek amal qiladi, faqat funksiya natijasi long turida bo'ladi. Agar hosil bo'lgan son qiymati long chegarasiga sig'masa, funksiya kutilmagan qiymatni qaytaradi.

`atof()`—belgi ko'rinishidagi haqiqiy sonni haqiqiy son ko'rinishiga o'tkazish funksiyasi.

Sintaksisi: `double atof(const char* ptr);` `ptr` ko'rsatgichdagi ASCIIz-satrn double turidagi suzuvchi nuqtali songa o'tkazishni amalga oshiradi.

`strtod()` funksiyasi `atof()` funksiyasidan farqli ravishda satrn double turidagi songa o'tkazishda konvertatsiya jarayoni uzulgan paytda aylantirish mumkin bo'lmagan birinchi belgi adresini ham qaytaradi. Bu o'z navbatida satrning xato qismini qayta ishlash imkoniyatini beradi.

Sintaksisi: `double strtod(const char *s, char **endptr);` `endptr` ko'rsatkichi konvertatsiya qilinishi mumkin bo'lmagan birinchi belgi adresi.

`itoa()` va `ltoa()` funksiyalari mos ravishda int va long turidagi sonlarni satr ko'rinishiga o'tkazadi. Bu funksiyalar mos ravishda quyidagi sintaksislarga ega: `char* itoa(int num, char *str, int radix);`

va `char* ltoa(long num, char *str, int radix);` `num` sonini `radix` argumentda ko'rsatilgan sanoq sistemasidagi ko'rinishini str satrda hosil qiladi.

Masala. O'nlik sanoq sistemasida berilgan 12345 sonini 2 lik, 8 lik, 16 lik sanoq sistemalariga o'tkazib, satr ko'rinishida tasvirlovchi dastur tuzilsin:

```
int main()
{
    char satr2[20],satr8[15],satr10[10],satr16[5];
    int son=12345;
    itoa(son,satr2,2);
    itoa(son,satr8,8);
    itoa(son,satr10,10);
    itoa(son,satr16,16);
    cout<<"Son ko'rinishlari"<<endl;
    cout<<"2 lik sanoq sistemasida :"<<satr2<<endl;
    cout<<"8 lik sanoq sistemasida :"<<satr8<<endl;
```

```
cout<<"10 lik sanoq sistemasida:"<<satr10<<endl;
cout<<"16 lik sanoq sistemasida:"<<satr16<<endl;
return 0;
}
```

Dastur ishlashi natijasida ekranga quyidagi satrlar chiqadi:

2 lik sanoq sistemasida :11000000111001;

8 lik sanoq sistemasida :30071;

10 lik sanoq sistemasida:12345;

16 lik sanoq sistemasida:3039;

Yuqorida ko'rib o'tilgan funksiyalar *char* turidagi massiv satri uchun o'rinli hisoblanadi.

C++ tilining 32 razryadli apparat platformalar uchun yaratilgan variantlarida, masalan, C++ Builder variantida standart satr turiga qo'shimcha sifatida string turi kiritilgan va u string sinfi ko'rinishida amalga oshirilgan. Bu turdagi satr uchun '\0' belgisi tugash belgisi hisoblanmaydi, ya'ni u oddiygina belgilar massivi sifatida qaraladi. string turidagi satr uzunligi, bajariladigan amallar natijasida dinamik ravishda o'zgarib turishi, uning tarkibida bir qator funksiyalar aniqlanganligi, bu tur bilan ishlashda ma'lum bir qulayliklar yaratadi.

string turidagi o'zgaruvchilar quyidagicha e'lon qilinadi:

```
string s1,s2,s3;
```

Bu turdagi satrlar uchun maxsus amallar va funksiyalar aniqlangan.

Satr turidagi o'zgaruvchiga qiymat berish:

```
string s1="bevosita qiymatlash";
```

```
string s2(s1);
```

```
string s3=s2;
```

string turidagi o'zgaruvchilar bilan ishlovchi funksiyalar:

Bir satr qismini boshqa satrga yuklash uchun quyidagi funksiyalarni ishlatish mumkin, ularning prototipi quyidagicha:

```
assign(const string &str);
```

```
assign(const string &str, unsigned int pos, unsigned int n);
```

```
assign(const char *str, int n);
```

Birinchi funksiya qiymat berish amali bilan ekvivalentdir.

Ikkinchi funksiya chaqiruvchi satrga argumentdagi str satrning pos o'rnidan n ta belgidan iborat bo'lgan satr qismini nusxalaydi. Agarda pos qiymati str satr uzunligidan katta bo'lsa, xatolik haqida ogohlantiriladi. Agar $pos+n$ ifodaning qiymati str satr uzunligidan katta bo'lsa, str satrining pos o'rnidan boshlab satr oxirigacha bo'lgan belgilar nusxalanadi. Bu qoida navbatdagi funksiyalar uchun o'rinlidir.

Uchinchi funksiya argumentdagi *char* turidagi *str* satrni *string* turiga aylantirib, funksiyani chaqiruvchi satrga qiymatlaydi.

Satr qismini boshqa satrga qo'shish funksiyalari:

append(const string &str);

append(const string&str,unsigned int pos,unsigned int n);

*append(const char *str,int n);*

Bu funksiyalarning yuqorida keltirilgan assign funksiyalaridan farqi shundaki, bu funksiyalar funksiyani chaqiruvchi satr oxiriga *str* satrining o'zini yoki uning qismini qo'shadi.

Satr qismini o'chirish funksiyalari:

erase(unsigned int pos=0,unsigned int n=pos);

Bu funksiya, uni chaqiruvchi satrning *pos* o'rnidan boshlab *n* ta belgini o'chirib tashlaydi. Agarda *pos* ko'rsatilmasa, satr boshidan boshlab o'chiriladi. Agar *n* ko'rsatilmasa, satrning oxirigacha bo'lgan belgilar o'chiriladi.

void clear() funksiyasi, uni chaqiruvchi satrni to'liq tozalaydi.

Bir satrga ikkinchi satr qismini joylashtiruvchi funksiyalar:

insert(unsigned int pos1,const string &str);

insert(unsigned int pos1,const string & str, unsigned int pos2, unsigned int n);

*insert(unsigned int pos1,const char *str, int n);*

Bu funksiyalar *append* kabi ishlaydi. Undan farqi shundaki, *str* satrini yoki uning qismini funksiyani chaqiruvchi satrning ko'rsatilgan *pos1* o'rnidan boshlab joylashtiradi. Bunda funksiyani chaqirgan satrning *pos1* o'rindan keyin joylashgan belgilar o'nga suriladi.

Bir satr qismining o'rniga boshqa satrning qismini qo'yish funksiyalari:

replace(unsigned int pos1,unsigned int n1,const string &str);

replace(unsigned int pos1, unsigned int n1,const string & str, unsigned int pos2,unsigned int n2);

*replace(unsigned int pos1, unsigned int n1,const char *str, int n);*

Bu funksiyalar *insert* kabi ishlaydi. Undan farqi shundaki, funksiya chaqiruvchi satrning ko'rsatilgan o'rnidan (*pos1*) *n1* ta belgilar o'rniga *str* satrini yoki uning *pos2* o'rindan boshlangan *n2* ta belgidan iborat qismini qo'yadi (almashtiradi).

Ikki satr o'zgaruvchisining qiymatlarini o'zaro almashtirish funksiyasi:

swap(string &str);

Satr qismini ajratib olish funksiyasi:

string substr(unsigned int pos=0, unsigned int n=npes) const;

Bu funksiya, uni chaqiruvchi satrning *pos* o'rnidan boshlab *n* ta belgini natija sifatida qaytaradi. Agarda *pos* ko'rsatilmasa, satr boshidan boshlab *n* ta belgi ajratib olinadi, agar *n* ko'rsatilmasa, *pos* dan boshlab satr oxirigacha bo'lgan belgilar natija sifatida qaytariladi.

string turidagi satrni *char* turiga o'tkazish funksiyalari:

*const char * c_str() const;*

bu funksiya *char* turdagi '\0' belgisi bilan tugaydigan satrga o'zgarmas ko'rsatkichni qaytaradi.

*const char * data() const;*

bu funksiya esa satr oxiriga '\0' belgisini qo'shmaydi.

Satr qismini izlash funksiyalari. Quyida ulardan asosiylarining tavsifini keltiramiz:

unsigned int find(const string &str, unsigned int pos=0) const;

bu funksiya, uni chaqirgan satrning ko'rsatilgan joyidan (*pos*) boshlab *str* satrni qidiradi va birinchi mos keluvchi satr qismining boshlanish o'rnini qiymat sifatida qaytaradi, aks holda maksimal musbat butun *npos* sonini qaytaradi, agar izlash o'rnini (*pos*) berilmasa, izlanayotgan qism satr satr boshidan boshlab izlanadi.

unsigned int find(char c, unsigned int pos=0) const;

bu funksiya oldingi funksiyadan farqli ravishda satrdan (*c*) belgini izlaydi.

unsigned int rfind(const string &str, unsigned int pos=npes) const;

bu funksiya, uni chaqirgan satrning ko'rsatilgan *pos* o'rnigacha *str* satrining birinchi uchragan joyining o'rnini qaytaradi, aks holda esa *npos* qiymatini qaytaradi, agar *pos* ko'rsatilmasa izlash satr oxirigacha davom ettiriladi.

Satr xossalarini aniqlash funksiyalari. *string* sinfida satr uzunligini, bo'shligini va uning egallagan xotira hajmini aniqlaydigan funksiyalar ham mavjud.

unsigned int size() const; //satr o'lchami;

unsigned int length() const; //satrdagi elementlar soni;

unsigned int max_size() const; //satrning maksimal uzunligi;

*unsigned int capacity() const; /*satrning xotiradan egallagan hajmini qaytaradi; */*

*bool empty() const; /*agar satr bo'sh bo'lsa, 1(ro'st), aks holda 0(yolg'on) qiymatini qaytaradi; */*

Satrlar bilan ishlovchi funksiyalar ko'p sonli bo'lib, yuqorida ularning nisbatan ko'p ishlatiladiganlaridan namunalar keltirildi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 5.010 Satr berilgan. Undagi raqamlar soni hisoblansin.
- 5.011 Satr berilgan. Uning belgilari teskari tartibda chop etilsin (telefon - > nofelet).
- 5.012 Bo'sh bo'lmagan satr berilgan. Satrda joylashgan belgilarning orasiga bittadan bo'sh joy qo'yishdan hosil bo'lgan satr chop etilsin.
- 5.013 Bo'sh bo'lmagan satr va $n(n>0)$ butun soni berilgan. Satrdagi belgilar orasiga n tadan "*" qo'yib, satr chop etilsin.
- 5.014 Satr berilgan. Undagi lotin alfavitining bosh harflari soni hisoblansin.
- 5.015 Satr berilgan. Satrga kirmagan barcha lotin harflarining soni hisoblansin.
- 5.016 Satr berilgan. Berilgan satrdagi barcha bosh harflar kichik harflarga aylantirilsin.
- 5.017 Satr berilgan. Berilgan satrdagi barcha kichik harflar bosh harflarga almashtirilsin.
- 5.018 Satr berilgan. Berilgan satrdagi barcha bosh harflar kichik harflarga, kichik harflar esa bosh harflarga almashtirilsin.
- 5.019 Satr berilgan. Agar satr butun sondan iborat bo'lsa 1, haqiqiy sondan iborat bo'lsa 2, satrni son ko'rinishiga o'tkazib bo'lmasa 0 chop etilsin.
- 5.020 Butun musbat son berilgan. Bu sonni tasvirlovchi raqamlardan iborat belgilar o'ngdan chapga qaragan tartibda chop etilsin.
- 5.021 Butun musbat sonni tasvirlovchi satr berilgan. Bu sonning raqamlari yig'indisi hisoblansin.
- 5.022 " $\langle raqam1 \rangle \pm \langle raqam2 \rangle \pm \dots \pm \langle raqamN \rangle$ " ushbu arifmetik ifodani tasvirlovchi satr berilgan. Satrdagi " $\pm$ " belgilar o'rniga, "+" yoki "-" amali qo'yilsin va ifodaning qiymati chiqarilsin.
- 5.023 Satrda butun musbat sonning ikkilik sanoq sistemasidagi ko'rinishi tasvirlangan. Bu sonning 10 lik sanoq sistemasidagi ko'rinishi chop etilsin.
- 5.024 Satrda butun musbat sonning 10 liksanoq sistemasidagi ko'rinishi tasvirlangan. Bu sonning 2 lik sanoq sistemasidagi ko'rinishi chop etilsin.
- 5.025 $n(n>0)$ butun soni va satr berilgan. n uzunlikka teng bo'lgan satr quyidagi ko'rinishda aniqlanadi: satr uzunligi n dan katta bo'lsa,

- uning o'ng tomonidan ortiqcha belgilar olib tashlansin, agar satr uzunligi n dan kichik bo'lsa, uning o'ng tomoniga nuqtalar qo'shilsin. Hosil qilingan satr chop etilsin.
- 5.026** Butun musbat n_1 , n_2 sonlar va 2 ta satr berilgan. Bu satrlardan foydalanib yangi satr hosil qilinsin: satrning dastlabki n_1 ta belgii ikkinchi satrning bosh qismidan, oxirgi n_2 ta belgi birinchi satrning oxirgi qismidan (chapdan o'ngga qarab) iborat bo'lsin.
- 5.027** Satr va ixtiyoriy belgi berilgan. Agar satrda berilgan belgi uchrasa, u ikkilantirilsin.
- 5.028** Ixtiyoriy belgi va 2 ta satr berilgan. Ikkinchi satrda berilgan belgi uchrasa uning oldiga birinchi satr joylashtirilsin.
- 5.029** Ixtiyoriy belgi va 2 ta satr berilgan. Birinchi satrda belgi uchrasa, shu belgi ortidan ikkinchi satr joylashtirilsin.
- 5.030** 2 ta satr berilgan. Agar birinchi satr ikkinchi satrda mavjud bo'lsa *1 (rost)* aks holda *0 (yolg'on)* qiymat chiqarilsin.
- 5.031** 2 ta satr berilgan. Ikkinchi satrda birinchi satrning necha marta uchrashi aniqlansin.
- 5.032** 2 ta satr berilgan. Ikkinchi satrdan birinchi satr bilan ustma-ust tushuvchi 1-qism satr o'chirilsin. Agar ikkinchi satrda birinchi satr topilmasa satr o'zgarishsiz chop etilsin.
- 5.033** 2 ta satr berilgan. Ikkinchi satrdan birinchi satr bilan ustma-ust tushuvchi oxirgi qism satr o'chirilsin. Agar s satrda *Birinchi* satr topilmasa s satr o'zgarishsiz chop etilsin.
- 5.034** 2 ta satr berilgan. Ikkinchi satrdan birinchi satr bilan ustma-ust tushuvchi barcha qism satrlar o'chirilsin. Agar ikkinchi satrda birinchi satr topilmasa ikkinchi satr o'zgarishsiz chop etilsin.
- 5.035** 3 ta satr berilgan. Birinchi satrdagi dastlabki uchragan ikkinchi qism satr uchinchi qism satr bilan almashtirilsin. Agar birinchi satrda ikkinchi satr topilmasa birinchi satr o'zgarishsiz chop etilsin.
- 5.036** 3 ta satr berilgan. Birinchi satrdagi oxirgi uchragan ikkinchi qism satr uchinchi qism satr bilan almashtirilsin. Agar birinchi satrda ikkinchi satr topilmasa birinchi satr o'zgarishsiz chop etilsin.
- 5.037** 3 ta satr berilgan. Birinchi satrda uchragan barcha ikkinchi qism satrlaruchinchi qism satr bilan almashtirilsin. Agar birinchi satrda ikkinchi satr topilmasa birinchi satr o'zgarishsiz chop etilsin.
- 5.038** Kamida 2 ta bo'sh joyga ega satr berilgan. Berilgan satrdagi 1- va 2- bo'sh joylar orasida joylashgan qism satr chiqarilsin.
- 5.039** Kamida 2 ta bo'sh joyga ega satr berilgan. Berilgan satrdagi 1- va

oxirgi bo'sh joylar orasida joylashgan qism satr chiqarilsin.

B toifa

- 5.040 Bo'sh joylar bilan ajratilgan so'zlardan tuzilgan satr berilgan. Satrdagi so'zlar soni topilsin.
- 5.041 Bosh harflar bilan terilgan va bo'sh joylar bilan ajratilgan so'zlardan iborat satr berilgan. 1- va oxirgi harflari bir xil bo'lgan so'zlar soni topilsin.
- 5.042 Bosh harflar bilan terilgan va bo'sh joylar bilan ajratilgan, so'zlardan iborat satr berilgan. Hech bo'lmaganda bitta "a" harfi bor bo'lgan so'zlar soni chop etilsin.
- 5.043 Bosh harflar bilan terilgan va bo'sh joylar bilan ajratilgan so'zlardan iborat satr berilgan. O'zida kamida 2 ta "a" harfini saqalovchi so'zlar soni chop etilsin.
- 5.044 Bo'sh joylar bilan ajratilgan, so'zlardan tuzilgan satr berilgan. Satrdagi eng qisqa so'zning uzunligi topilsin.
- 5.045 Bo'sh joylar bilan ajratilgan, so'zlardan tuzilgan satr berilgan. Satrdagi eng uzun so'zning uzunligi topilsin.
- 5.046 Bosh harflar bilan terilgan va bo'sh joylar bilan ajratilgan, so'zlardan iborat satr berilgan. Satrdagi har bir so'zning 1- harfi uning qolgan qismida uchrasa, ular "." bilan almashtirilsin.
- 5.047 Bosh harflar bilan terilgan va bo'sh joylar bilan ajratilgan, so'zlardan iborat satr berilgan. Satrdagi har bir so'zning oxirgi harfi uning qolgan qismida uchrasa, ular "." bilan almashtirilsin.
- 5.048 Bo'sh joylar bilan ajratilgan, so'zlardan tuzilgan satr berilgan. Satrdagi so'zlarning har biri teskari tartibda joylashtirilib, chop etilsin.
- 5.049 Bosh harflar bilan terilgan va bo'sh joylar bilan ajratilgan, so'zlardan iborat satr berilgan. Satrdagi so'zlar alfavit tartibida joylashtirilib chop etilsin.
- 5.050 Satrda gap joylashgan. Gapdagi har bir so'zning birinchi harfi bosh harfga o'tkazilsin.
- 5.051 Satrda gap joylashgan. Gapdagi tinish belgilar soni hisoblansin.
- 5.052 Satrda gap joylashgan. Gapdagi unli harflar soni hisoblansin.
- 5.053 Satrda gap joylashgan. Gapdagi eng uzun so'z topilsin. Agar bunday so'z bir nechta bo'lsa, ulardan 1-si chiqarilsin. So'zdagi belgilar sonini hisoblashda tinish belgilari, chegaralovchi bo'sh joylar hisobga olinmasin.

- 5.054** Satrda gap joylashgan. Gapdagi eng qisqa soʻz topilsin. Agar bunday soʻz bir nechta boʻlsa, ulardan 1-si chiqarilsin. Soʻzdagi belgilar sonini hisoblashda tinish belgilari, chegaralovchi boʻsh joylar hisobga olinmasin.
- 5.055** Satrdagi gap soʻzlari bir nechta boʻsh joylar bilan ajratilgan. Gap, soʻzlari orasida bittadan boʻsh joy qoldirib tasvirlansin.
- 5.056** Satrda disk nomi, kataloglar roʻyxati, faylning xususiy nomi va kengaytmasi, yaʼni faylning toʻliq nomi berilgan. Bu satrdan faylning xususiy nomi (kengaytmasisiz) ajratib olinsin.
- 5.057** Satrda disk nomi, kataloglar roʻyxati, faylning xususiy nomi va kengaytmasi, yaʼni faylning toʻliq nomi berilgan. Bu satrdan faylning kengaytmasi (ajratuvchi nuqtadan tashqari) ajratib olinsin.
- 5.058** Satrda faylning toʻliq nomi berilgan. Ushbu satrdan 1-katalog nomi ajratib olinsin. Agar fayl ildiz katalogida joylashgan boʻlsa, ekranga “\” belgisi chiqarilsin.
- 5.059** Satrda faylning toʻliq nomi berilgan. Ushbu satrdan oxirgi katalog nomi ajratib olinsin. Agar fayl ildiz katalogida joylashgan boʻlsa, ekranga “\” belgisi chiqarilsin.
- 5.060** Satrda gap joylashgan. Gap quyidagi qoida boʻyicha shifrlansin: har bir harf alfavitdagi oʻzidan keyin keluvchi harf bilan almashtirilsin. Agar harf alfavitning eng oxirgi kichik harfi boʻlsa, u katta “A” harfiga almashtirilsin.
- 5.061** k butun ($0 < k < 10$) soni va kichik harflardan tashkil topgan gap berilgan. Gapdagi harflar alfavitda joylashgan oʻrnidan k ta keyin turgan harf bilan almashtirilsin. Boʻsh joy, tinish belgilar va maxsus belgilar oʻzgarishsiz qoldirilsin. Agar harf alfavitning oxirgi k ta elementlaridan biri boʻlsa, u “harf kodi”+ $k-26$ oʻrindagi harf bilan almashtirilsin.
- 5.062** k butun ($0 < k < 10$) soni va kichik harflardan tashkil topgan gap berilgan. Gapdagi harflar alfavitda joylashgan oʻrnidan k ta oldin turgan harf bilan almashtirilsin. Boʻsh joy, tinish belgilar va maxsus belgilar oʻzgarishsiz qoldirilsin. Agar harf alfavitning dastlabki k ta elementlaridan biri boʻlsa, u “harf kodi”- $k+26$ oʻrindagi harf bilan almashtirilsin.
- 5.063** Gap joylashgan satr berilgan. Gapda joylashgan barcha belgilarning juft oʻrindagilari ajratib olinsin, qolgan (toq oʻrindagilari) belgilar teskari tartibda joylashtirilib, juft oʻrindagi belgilardan tuzilgan satr boʻlagi davomidan yozilsin.

- 5.064** Gapda joylashgan barcha belgilarning juft o'rindagilari ajratib olinib, qolgan (toq o'rindagilari) belgilar teskari tartibda joylashtirilib, juft o'rindagi belgilardan tuzilgan satr bo'lagi davomidan yozilgan satr berilgan. Bu satr asl holiga o'tkazilsin.
- 5.065** Satrda raqamlar va kichik lotin harflar joylashgan. Agar satrdagi harflar alfavit tartibida joylashgan bo'lsa 0, qolgan hollarda tartibni buzuvchi 1-belgining *ASCII* kodi chiqarilsin.
- 5.066** Lotin harflari va oddiy qavslardan iborat satr berilgan. Agar qavslar to'g'ri qo'llanilgan bo'lsa 0, aks holda xatolik ro'y bergan pozitsiya nomeri chop etilsin.
- 5.067** Lotin harflari va "()", "[]", "{}" ko'rinishdagi qavslar joylashgan satr berilgan. Agar qavslar to'g'ri qo'llanilgan bo'lsa 0, aks holda xatolik ro'y bergan pozitsiya nomeri chop etilsin.

5.3. Bobga doir qiyinroq masalalar

5.068

Qism satr.

Sizning firmangiz xujjat matnini arxivlovchi dasturiy ta'minot ustida ishlamoqda. Sizga dasturiy ta'minotga tegishli funksiya yaratish vazifasi topshirildi. Bu funksiya matnda takrorlanuvchi eng uzun, kesishmaydigan qism satrni topishi kerak. (Registrni hisobga olinish lozim. Ikki qism satr turli hisoblanadi, bu satrlardagi mos o'rindagi harflar katta kichikligi va turli ekanligi bilan farq qilsa.) Agar bunday qism satrlar bir nechta bo'lsa eng birinchi kelgani olinishi lozim.

Masalan, "ABCDABCFG", satrida eng uzun qism satr "ABC". Masalan, "ABABA", satrda "AB" va "BA" qism satrlar qaytariladi, lekin "AB", tanlanadi, chunki u oldin keladi. ("ABA" qism satr ikki marta qaytariladi, lekin o'zaro kesishadi, shuning uchun olinmaydi.

Berilganlar:

Satr simvollari soni 1 dan 50 gacha bo'lib, "A"- "Z", "a"- "z", "0"- "9" belgilar va bo'sh joylardan iborat.

Natija:

Qism satr chiqarilsin. Agar bunday qism satr mavjud bo'lmasa bo'sh joy chiqarilsin.

Berilganlar:	Natija:
"ABCDABCFG"	"ABC"
"ABABA"	"AB"
"This is a test."	"is "
"Testing testing 1 2 3."	"esting "
"The quick brown fox jumps over the lazy dog."	"he "

5.069

Polyakcha struktura [11].

Postfiks strukturasida (yoki polyakcha strukturaning teskarisi) amallar ikkita operanddan keyin yoziladi. Masalan, ikkita A va B sonlar yig'indisi $AB+$ ko'rinishida yoziladi. Shuningdek $BC+D*$ strukturasini bizga yaxshi ma'lum bo'lgan $(B+C)*D$ strukturasini anglatadi, $ABC+D*+$ strukturasini esa $A+(B+C)*D$ ni anglatadi. Postfiks strukturaning yaxshi tomoni shundan iboratki, u qavs ishlatish va operatorlarning o'qilishiga qo'shimcha shartlar talab qilmaydi. Teskari polyakcha strukturada ifoda berilgan. Uning qiymatini toping.

Berilganlar:

Bitta satrda postfiks strukturadagi ifoda berilgan. Unda sonlar va $+$, $-$, $*$ amallari mavjud. Qatorda 100 dan ortiq bo'lmagan sonlar va amallar berilgan. Sonlar va amallar bo'sh joy bilan ajratilgan.

Natija:

Yozilgan ifodaning qiymati topilsin. Ifodani hisoblash natijalari va oraliq hisob-kitoblar natijalari modul boyicha 231 dan kichik.

Berilganlar:	Natija:
8 9 + 1 7 - *	-102

5.070

Invers [12].

Sizga n butun soni berilgan. Sizning vazifangiz uning ikkilik kodini 31 razryadga to'ldirib (0 bilan), undagi har bir bitni teskari qiymatga o'zgartirib, (0 bo'lsa 1 ga, 1 bo'lsa 0 ga) qaytadan 10 lik sanoq sistemasida chiqarishdan iborat.

Berilganlar:

Birinchi qatorda n soni berilgan ($0 \leq n \leq 231-1$).

Natija:

Birinchi qatorda masalaning javobini chiqaring.

N_{q}	Berilganlar:	Natija:
1	451457123	1696026524

2	0	2147483647	
---	---	------------	--

Izoh: 1-misolda 451457123 sonini ikkilikga o'tirib 31 razryadga to'ldirilgandan keyin 0011010111010001011000001100011, har bir bitini teskari aylantirgandan so'ng 1100101000101110100111110011100 va u o'nlikda 1696026524 ga teng.

VI Bob. Funksiyalar

Dasturlashga tizimli yondoshuv shundan iboratki, dastur tuzuvchi oldiga qo'yilgan masala oldindan ikkita, uchta va undan ortiq nisbatan kichik qism masalalarga bo'linadi. O'z navbatida bu qism masalalar ham yanada kichik qism masalalarga bo'linishi mumkin. Bu jarayon kichik hajmdagi masalalarni oddiy standart amallar yordamida yechish mumkin bo'lguncha davom etadi. Shu yo'l bilan qo'yilgan masalani dekompozitsiyalash amalga oshiriladi.

Bundan tashqari, dasturlashda shunday holatlar kuzatiladi, ya'ni dasturning turli joylarida mazmunan bir xil algoritmlarni bajarishga to'g'ri keladi. Algoritmning bu bo'laklari asosiy yechilayotgan masaladan ajratib olingan biror qism masalani yechishga mo'ljallangan bo'lib, mustaqil qiymatga (natijaga) ega bo'ladi.

C++ tilida dastur bo'lagini alohida ajratib olib, uni funksiya ko'rinishida ifodalash imkoniyati mavjud.

Dasturlash tillarida *funksiya* – masala yechishdagi kalit elementlardan biridir.

Dasturda ishlatiladigan har qanday funksiya e'lon qilinishi kerak. Odatda funksiyalar e'loni sarlavha fayllarda e'lon qilinadi va *#include* direktivasi yordamida dastur matniga qo'shiladi. Funksiya e'lonini funksiya prototipi tavsiflaydi. Funksiya prototipi quyidagi ko'rinishda bo'ladi:

<qaytariluvchi qiymat turi> <funksiya nomi>(<parametrlar ro'yxati >);

bu yerda *<qaytariluvchi qiymat turi>*—funksiya ishlashi natijasida qaytariladigan qiymatning turidir. Agar qaytariladigan qiymat turi ko'rsatilmagan bo'lsa, kelishuv bo'yicha funksiya qaytaradigan qiymat turi *int* deb hisoblanadi, *<parametrlar ro'yxati>*—vergul bilan ajratilgan funksiya parametrlarining turi va nomlari ro'yxati. Parametr nomini yozmasa ham bo'ladi. Ro'yxat bo'sh bo'lishi ham mumkin. Funksiya prototiplariga misollar:

int almashtir(int,int);

double minimum(double a, double b);

void fakt();

void bosmalash(void);

Agar dastur matnida funksiya aniqlanishi, uni chaqiradigan funksiyalar matnidan oldin yozilgan bo'lsa, bunday funksiya prototipi tushirib qoldirilishi ham mumkin. Ammo bu holat yaxshi usul

hisoblanmaydi, ayniqsa o‘zaro bir-biriga murojaat qiluvchi funksiyalarni e‘lon qilishda muammolar yuzaga keladi.

Funksiya matni–funksiya sarlavhasi va figurali qavsga (‘{’,’}’) olingan, amaliy mazmunga ega tanadan iborat bo‘ladi. Agar funksiyaning turi *void* turidan farqli bo‘lsa, uning tanasida albatta mos turdagi parametrga ega *return* operatori bo‘lishi shart. Funksiya tanasida bittadan ortiq *return* operatori bo‘lishi ham mumkin. Ularning ixtiyoriy birortasi bajarilganda funksiyadan chiqib ketish amalga oshadi. Agar funksiyaning qiymati dasturda ishlatilmaydigan bo‘lsa, funksiyadan chiqish uchun parametrsiz *return* operatori ishlatiladi yoki umuman *return* ishlatilmasligi ham mumkin. Oxirgi holatda funksiyadan chiqish, funktsiyani yakunlovchi figurali qavsga yetib kelinganda amalga oshadi.

Funksiya dasturning biror modulida faqat bir marta aniqlanishi kerak, uning e‘loni esa funktsiyani ishlatadigan modullarda bir necha marta yozilishi mumkin. Funksiya aniqlanishida sarlavhadagi barcha parametrlar nomlari yozilishi shart.

Funksiya dasturga ma‘lum bir ishni amalga oshirish uchun chaqiriladi. Funksiyaga murojaat qilinganda, u o‘ziga yuklangan masalani yechadi va o‘z ishini tugatib, kutilgan qiymatni natija sifatida qaytaradi.

Funktsiyani chaqirish uchun uning nomi va undan keyin qavs ichida argumentlar ro‘yxati beriladi:

<funksiya nomi>(<argument₁>,<argument₂>,...,<argument_n>);

bu yerda har bir *<argument>*, funksiya tanasiga uzatiladigan va keyinchalik hisoblash jarayonida ishlatiladigan o‘zgaruvchi, ifoda yoki o‘zgarmasdir. Argumentlar ro‘yxati bo‘sh bo‘lishi ham mumkin.

Funktsiyalar o‘z tanasida boshqa funksiyalarni, hatto o‘zini ham chaqira oladi. O‘z tanasida o‘zini chaqiradigan funksiyalarga *rekursiv* funksiyalar deyiladi.

Funksiya prototipida yoki aniqlanish sarlavhasida ko‘rsatilgan parametrlar *formal* parametrlar deyiladi, funksiya chaqirilishida ko‘rsatilgan argumentlarga *faktik* parametrlar deyiladi.

C++ tilidagi har bir dasturda albatta *main()*–bosh funksiya bo‘lishi shart. Aynan shu funksiya yuklagich tomonidan chaqirilishi bilanoq dasturning bajarilishi boshlanadi.

Aksariyat hollarda *main()* funksiyasining parametrlar ro‘yxati bo‘sh bo‘ladi. Agar dastur ishga tushirilayotganda, uning buyruq satrida ma‘lum bir parametrlarni uzatish zarur bo‘lsa, *main()* funksiyasining sintaksisi o‘zgaradi:

```
int main(int argc, char* argv[]);
```

Bu yerda *argc* – uzatiladigan parametrlar soni, *argv[]*- bir-biridan punktuatsiya belgilari (yoki probel) bilan ajratilgan parametrlar ro'yxatini o'z ichiga olgan massivga ko'rsatkich.

Quyida dasturchi tomonidan tuzilishi mumkin bo'lgan bir qator funksiyalar va ularning tuzilishi bilan tanishamiz.

6.1. Sonli parametrli funksiyalar

Funksiyalarning dasturda tasvirlanishini namoyish etish maqsadida son kvadratini hisoblash masalasi uchun dasturchi funksiyasini tuzamiz.

Buning uchun funksiya prototipini *funksiya.h* sarlavha fayliga joylashtiramiz:

```
long son_kv(int);
```

Asosiy dasturga ushbu sarlavha faylini qo'shish orqali *son_kv()* funksiya e'loni dastur matniga kiritiladi:

```
#include <iostream.h>
#include "funksiya.h"
int main()
{
    int butun_uzgaruvchi=6;
    cout<<son_kv(butun_uzgaruvchi);
    return 0;
}
long son_kv(int x)
{
    return x*x;
}
```

Xuddi shu masalani sarlavha faylidan foydalanmagan holda, funksiya e'lonini dastur matniga yozish orqali ham hal etish mumkin:

```
#include <iostream.h>
long son_kv(int);
int main()
{
    int butun_uzgaruvchi=6;
    cout<<son_kv(butun_uzgaruvchi);
    return 0;
}
long son_kv(int x)
```



```
{  
    return x*x;  
}
```

Har ikkala dastur bir xil ishlaydi va natija sifatida ekranga 36 soni chop etiladi.

C++ tilida funksiya chaqirilishida argumentlarni tushirib qoldirish imkoniyati ham mavjud. Bu holatga funksiya prototipida ushbu parametrlarning kelishuv bo'yicha qiymatini ko'rsatish orqali erishiladi.

Buni quyidagi misolda ko'ramiz:

```
//funksiya prototipi  
void Butun_son(int i, bool b=true, char belgi='\n');  
//funksiyaning chaqirish variantlari:  
Butun_son(5,false,'h');  
Butun_son(8,false);  
Butun_son(74);
```

Birinchi chaqiruvda barcha parametrlar mos argumentlar orqali qiymatlarni, ikkinchi holda esa *i* parametr 8 qiymatini, *b* parametri esa '*false*' qiymatini va belgi o'zgaruvchisi kelishuv bo'yicha '\n' qiymatini qabul qiladi. Uchinchi holatda esa *i* o'zgaruvchisi 74 qiymatini, *b* va *belgi* o'zgaruvchilari mos ravishda kelishuv bo'yicha *true* va '\n' qiymatlarini qabul qiladi.

Kelishuv bo'yicha qiymat berishning bitta sharti bor – parametrlar ro'yxatida kelishuv bo'yicha qiymat berilgan parametrdan keyingi barcha parametrlar kelishuv bo'yicha qiymatga ega bo'lishi shart.

Masala 1. Haqiqiy sonni ko'rsatilgan aniqlikda chop etuvchi dastur tuzilsin.

Qo'yilgan masalani yechishda sonni darajaga oshirish funksiyasi – *pow()* va suzuvchi nuqtali uzun sondan modul olish *fabs()* funksiyasidan foydalaniladi. Bu funksiyalar prototipi <math.h> sarlavha faylida joylashgan.

```
#include <iostream.h>  
#include <math.h>  
void chop_etish (double son, double aniqlik=1, bool b=true);  
int main()  
{ double pi1=-3.141592654;  
  chop_etish(pi1,4,false);  
  chop_etish(pi1,2);  
  chop_etish(pi1);  
  return 0; }
```

```
void chop_etish(double son, double aniqlik=1, bool b=true)
{ if(!b) son=fabs(son);
  son=(int)(son*pow(10,aniqlik));
  son=son/pow(10,aniqlik);
  cout<<son<<'\\n'; }
```

Dasturda sonni turli aniqlikda (aniqlik parametri qiymati orqali) chop etish uchun har xil variantlarda *chop_etish()* funksiyasi chaqirilgan. Dastur ishlashi natijasida ekranda quyidagi sonlar chop etiladi:

3.1415

-3.14

-3.1

Parametrning kelishuv bo'yicha beriladigan qiymati o'zgarmas, global o'zgaruvchi yoki biror boshqa funksiya qaytaradigan qiymat ham bo'lishi mumkin.

Odatda o'zgaruvchilar funksiya tanasida yoki undan tashqarida e'lon qilinadi. Funksiya tanasida e'lon qilingan o'zgaruvchilarga *lokal* o'zgaruvchilar deyiladi. Bunday o'zgaruvchilar xotiradagi dastur stekida joylashadi va ular faqat o'zlari e'lon qilingan funksiya tanasida amal qiladi. Boshqaruv asosiy funksiyaga qaytishi bilan *lokal* o'zgaruvchilar uchun ajratilgan xotira bo'shatiladi.

Global o'zgaruvchilar dastur matnida funksiyalar aniqlanishidan oldin e'lon qilinadi va ular e'lon qilingan joyidan boshlab dastur oxirigacha amal qiladi.

Kompilyator ishlashi natijasida har bir funksiya mashina kodi ko'rinishiga o'tadi. Agar dasturda funktsiyani chaqirish ko'rsatmasi bo'lsa, shu joyda funktsiyani adresi bo'yicha chaqirishning mashina kodi shakllanadi. Funksiyani chaqirish protsessor tomonidan qo'shimcha vaqt va xotira resurslarini talab qiladi. Chaqiriladigan funksiya hajmi unchalik katta bo'lmagan hollarda, funktsiyani chaqirish kodi o'rniga kompilyatorga funksiya tanasining o'zini joylashtirish maqsadga muvofiq bo'ladi. Bu ish funksiya prototipini inline kalit so'zi bilan e'lon qilish orqali amalga oshiriladi. Natijada hajmi oshgan, lekin nisbatan tez bajariladigan dastur kodi yuzaga keladi.

Ayrim algoritmlar berilganlarning har xil turdagi qiymatlari uchun qo'llanilishi mumkin. Masalan, ikkita sonning maksimumini topish algoritmidan berilgan sonlar butun yoki haqiqiy turda bo'ladi. Bunday hollarda bu algoritmlarni amalga oshiradigan funksiyalarning nomlari bir xil bo'lishi ma'qul. Bir nechta funktsiyani bir xil nomlash, lekin har xil turdagi parametrlar bilan ishlatish funktsiyani qayta yuklash deyiladi.

Kompilyator parametrlar turiga va soniga qarab mos funksiyani chaqiradi. Bunday amal «*hal qilish amali*» deyiladi va uning maqsadi parametrlarga aynan mos keladigan funksiyani chaqirishdir. Agar bunday funksiya topilmasa kompilyator xatolik haqida xabar beradi. Funksiyani aniqlashda funksiya qaytaruvchi qiymat turining ahamiyati yo‘q.

Masala 2. Berilgan sonlarning kattasini topish dasturi tuzilsin.

```
#include <iostream.h>
int max(int, int);
char max(char, char);
float max (float, float);
int max (int, int, int);
void main()
{
    int a,b,k; char c,d; float x,y;
    cin>>a>>b>>k>>c>>d>>x>>y;
    cout<<max(a,b)<<max(c,d)<<max(a,b,k)<<max(x,y);
}
int max(int i, int j) {return (i>j) ?i:j;}
char max(char s1, char s2) {return (s1>s2) ?s1:s2;}
float max (float x, float y) {return (x>y)?x:y;}
int max (int i, int j, int k)
{return (i>j) ?(i>k?i:k;) (j>k)?j:k;}
```

Agar funksiya chaqirilishida argument turi uning prototipidagi xuddi shu o‘rindagi parametr turiga mos kelmasa, kompilyator uni parametr turiga keltirilishga harakat qiladi. Masalan, bool va char turlarini int turiga, float turini double turiga va int turini double turiga o‘tkazishga.

Qayta yuklanuvchi funksiyalardan foydalanishda quyidagi qoidalarga rioya qilish kerak:

- qayta yuklanuvchi funksiyalar bitta ko‘rinish sohasida bo‘lishi kerak;

- qayta yuklanuvchi funksiyalarda kelishuv bo‘yicha parametrlar ishlatilsa, bunday parametrlar barcha qayta yuklanuvchi funksiyalarda ham ishlatilishi va ular bir xil qiymatga ega bo‘lishi shart;

- agar funksiyalar parametrlarining turi faqat “*const*” va ‘&’ belgilari bilan farq qiladigan bo‘lsa, bu funksiyalar qayta yuklanmaydi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 6.001** Nol bo'lmagan haqiqiy a va b sonlari ustida 1 ta arifmetik amal bajaruvchi haqiqiy turga tegishli $Calc(a,b,op)$ nomli funksiya tasvirlansin. Bu yerda op parametri 1 bo'lsa "ayirish", 2 bo'lsa "ko'paytirish", 3 bo'lsa "bo'lish", boshqa hollarda "qo'shish" amaliga ekvivalent hisoblanadi. Bu funksiya foydalanib, berilgan k va l sonlari uchun n_1, n_2, n_3 operatsiyalardagi qiymatlar chop etilsin.
- 6.002** Koordinata o'qlarida yotmaydigan (tekislikdagi) nuqtaning qaysi chorakda joylashganligini aniqlovchi butun turga tegishli $Quarter(x,y)$ nomli funksiya tasvirlansin. Bu funksiya foydalanib, berilgan 3 ta koordinata o'qlarida yotmaydigan nuqtalarning qaysi choraklarda joylashganligi aniqlansin.
- 6.003** Agar berilgan butun son juft bo'lsa " 1 (rost)" aks holda " 0 (yolg'on)" qiymat qaytaruvchi mantiqiy turga tegishli $Even(k)$ nomli funksiya tasvirlansin. Bu funksiya foydalanib, berilgan 10 ta butun son dan iborat ketma ketlikdagi juft sonlarning miqdori topilsin.
- 6.004** Berilgan butun turga tegishli $k(k>0)$ parametr, biror butun sonning kvadratiga teng bo'lsa " 1 (rost)" aks holda " 0 (yolg'on)" qiymat qaytaruvchi mantiqiy turga tegishli $IsSquare(k)$ nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan 10 ta butun son dan iborat ketma ketlikdagi to'la kvadrat bo'lgan sonlar miqdori aniqlansin.
- 6.005** Berilgan butun k ($k>0$) parametr, 5 ning biror darajasiga teng bo'lsa 1 (rost) aks holda 0 (yolg'on) qiymatini qaytaruvchi mantiqiy turga tegishli $IsPowerS(k)$ funksiyasi tasvirlansin. Bu funksiya foydalanib berilgan 10 ta butun son dan iborat ketma ketlikdagi 5 ning darajalariga teng bo'lgan sonlarning miqdori topilsin.
- 6.006** Berilgan butun k ($k>0$) parametr n ($n>1$) ning biror darajasiga teng bo'lsa 1 (rost) aks holda 0 (yolg'on) qiymat qaytaruvchi mantiqiy turga tegishli $IsPowerN(k,n)$ nomli funksiyasi tasvirlansin. Bu funksiya foydalanib berilgan 10 ta butun son dan iborat ketma ketlikdagi n ning darajalariga teng bo'lgan sonlarning miqdori topilsin.
- 6.007** n ($n>1$) parametr tub son bo'lsa 1 (rost), aks holda 0 (yolg'on)

qiymat qaytaruvchi mantiqiy turga tegishli $IsPrime(n)$ nomli funksiya tasvirlansin. Bu funksiya dan foydalanib, berilgan 10 ta sondan iborat ketma ketlikdagi tub sonlar miqdori aniqlansin.

- 6.008** Berilgan k butun musbat sondagi raqamlar miqdorini aniqlovchi butun turga tegishli $Digit\ Count(k)$ nomli funksiya tasvirlansin. Bu funksiya dan foydalanib berilgan 5 ta musbat butun sonning har biridagi raqamlari soni aniqlansin.

B toifa

- 6.009** Butun turga tegishli k parametr *palindrom* bo'lsa 1 (rost) aks holda 0 (*yolg'on*) qiymat qaytaradigan mantiqiy turga tegishli $Ispalindron(K)$ nomli funksiya tasvirlansin. Funksiyani tasvirlashda $Digit\ count$ va $Digit\ N$ funksiyalaridan foydalanish mumkin. Bu funksiya dan foydalanib berilgan 5 ta butun musbat sondan iborat ketma ketlikdagi palendrom sonlar miqdori aniqlansin. (*palendrom son – o'ng va chapdan bir xil o'qiladigan son*).

- 6.010** Agar burchak o'lchovi gradusda berilgan bo'lsa uni radianda ifodalovchi haqiqiy turga tegishli $DegToRad(d)$ nomli funksiya tasvirlansin(d haqiqiy son $0 < d < 360$). Bu funksiya dan foydalanib graduslarda berilgan 4 ta burchak o'lchovlarining har biri uchun radian qiymatlari aniqlansin.

- 6.011** Agar burchak o'lchovi radianda berilgan bo'lsa uni gradusda ifodalovchi haqiqiy turga tegishli $RadtoDeg(r)$ nomli funksiya tasvirlansin(r haqiqiy son $0 < r < 2p$). Bu funksiya dan foydalanib, radianlarda berilgan 4 ta burchak o'lchovlarining har biri uchun gradus qiymatlari aniqlansin.

- 6.012** n faktorialni hisoblovchi haqiqiy turga tegishli $Fact(n)$ nomli funksiya tasvirlansin. Bu funksiya dan foydalanib, berilgan 5 ta butun musbat sonning har biri uchun faktoriallar hisoblansin.

- 6.013** $n!!$ ni hisoblovchi haqiqiy turga tegishli $Fact2(n)$ nomli funksiya tasvirlansin.

$n!!$ bu

agar n toq bo'lsa $n!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot n$

agar n juft bo'lsa $n!! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot n$

Bu funksiya dan foydalanib, berilgan 5 ta butun musbat sonlarning har biri uchun

$n!!$ lar hisoblansin.

- 6.014** f_k Fibonachchi sonlarining n -hadini hisoblaydigan butun turga

tegishli $Fib(n)$ nomli funksiya tasvirlansin. Bu funksiya foydalanib $n_1, n_2, \dots, n_5$ nomlarga to'g'ri keluvchi Fibonachchi sonlari topilsin.

- 6.015** $a^b = e^{b \ln a}$ dan foydalanib a sonining b – darajasini hisoblovchi haqiqiy turga tegishli $Power1(a,b)$ nomli funksiya tasvirlansin. (a va b – haqiqiy turga tegishli) a ning qiymati 0 yoki manfiy bo'lsa funksiya 0 qiymatni qaytarsin. Bu funksiya foydalanib, p, a, b, c sonlari berilganda a^p, b^p, c^p darajalari hisoblansin.

- 6.016** e^x funksiya
$$e^x = 1 + x + \frac{x^2}{(2!)} + \frac{x^3}{(3!)} + \dots + \frac{x^n}{(n!)} \dots$$

(bu yerda $n=1,2,\dots$) formula asosida taqribiy hisoblovchi haqiqiy turga tegishli $Exp1(x,eps)$ (x va eps ($eps>0$) haqiqiy turga tegishli sonlar) nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan x uchun eps ($eps>0$) ning 3 ta turli qiymatlarida e^x ning qiymatlari hisoblansin. Bu yerda eps aniqlik darajasini belgilovchi yetarlicha kichik son bo'lib, berilgan ketma ketlikning n - va $(n-1)$ - hadlari ayirmasining modulidan kichik yoki teng bo'lish sharti bilan aniqlanadi.

- 6.017** $\sin(x)$ ning qiymatini
$$\sin(x) = x - \frac{x^3}{(3!)} + \frac{x^5}{(5!)} - \dots + \frac{(-1)^n \cdot x^{2n+1}}{((2 \cdot n + 1)!)} + \dots$$

(bu yerda $n=1,2,\dots$) formula asosida taqribiy hisoblovchi haqiqiy turga tegishli $\sin1(x,eps)$ (x va eps ($eps>0$) haqiqiy turga tegishli sonlar) nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan x uchun eps ning 3 ta turli qiymatlarida $\sin(x)$ ning qiymati hisoblansin. Bu yerda eps aniqlik darajasini belgilovchi yetarlicha kichik son bo'lib, berilgan ketma ketlikning n - va $(n-1)$ - hadlari ayirmasining modulidan kichik yoki teng bo'lish sharti bilan aniqlanadi.

- 6.018** $\cos(x)$ ning qiymatini

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^n \cdot x^{2n}}{(2 \cdot n)!} + \dots$$

(bu yerda $n=1,2,\dots$) formula asosida taqribiy hisoblovchi haqiqiy turga tegishli $\cos1(x,eps)$ (x va eps ($eps>0$) haqiqiy turga tegishli sonlar) nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan x uchun eps ning 3 ta turli qiymatlarida $\cos(x)$ ning qiymati hisoblansin. Bu yerda eps aniqlik darajasini belgilovchi yetarlicha kichik son bo'lib,

berilgan ketma ketlikning n - va $(n-1)$ - hadlari ayirmasining modulidan kichik yoki teng bo'lish sharti bilan aniqlanadi.

6.019 $\ln(x)$ ning qiymatini

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^n \cdot x^{n+1}}{(n+1)} + \dots$$

(bu yerda $n=1,2,\dots$) formula asosida taqribiy hisoblovchi haqiqiy turga tegishli $\ln I(x, eps)$ (x va $eps(eps>0)$ haqiqiy turga tegishli sonlar) nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan x uchun e ning 3 ta turli qiymatlarida $\ln(1+x)$ ning qiymati hisoblansin. Bu yerda eps aniqlik darajasini belgilovchi yetarlicha kichik son bo'lib, berilgan ketma ketlikning n - va $(n-1)$ - hadlari ayirmasining modulidan kichik yoki teng bo'lish sharti bilan aniqlanadi.

6.020 $\arctg(x)$ ning qiymatini

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + \frac{(-1)^n \cdot x^{2n+1}}{2n+1} + \dots$$

(bu yerda $n=1,2,\dots$) formula asosida taqribiy hisoblovchi haqiqiy turga tegishli $\arctg I(x, eps)$ (x va $eps(eps>0)$ haqiqiy turga tegishli sonlar) nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan x uchun eps ning 3 ta turli qiymatlarida $\arctg(x)$ ning qiymati hisoblansin. Bu yerda eps aniqlik darajasini belgilovchi yetarlicha kichik son bo'lib, berilgan ketma ketlikning n - va $(n-1)$ - hadlari ayirmasining modulidan kichik yoki teng bo'lish sharti bilan aniqlanadi.

6.021 $(1+x)^a$ ning qiymatini

$$(1+x)^a = 1 + a \cdot x + \frac{a \cdot (a-1) \cdot x^2}{2!} + a \cdot (a-1) \cdot \dots \cdot (a-n+1) \cdot \frac{x^n}{n!} + \dots$$

(bu yerda $n=1,2,\dots$) formula yordamida hisoblovchi haqiqiy turga tegishli $Power4(x, a, eps)$ (x, a va $eps(eps>0)$ haqiqiy turga tegishli sonlar) nomli funksiya tasvirlansin. Bu funksiya foydalanib berilgan x va a ning qiymatlari uchun eps ning 3 ta turli qiymatlarida $(1+x)^a$ ning qiymati hisoblansin. Bu yerda eps aniqlik darajasini belgilovchi yetarlicha kichik son bo'lib, berilgan ketma-ketlikning n - va $(n-1)$ - hadlari ayirmasining modulidan kichik yoki teng bo'lish sharti bilan aniqlanadi.

6.022 Evklid algoritmidan foydalanib 2 ta a va b musbat butun sonlari uchun eng katta umumiy bo'luvchini topadigan (EKUB)

- $NOD2(a,b)$ nomli funksiya tasvirlansin. $NOD(a,b)=NOD(b,a\%b)$, agar $b\neq 0$; $NOD(a,0)=a$. Bu funksiya dan foydalanib a, b, c, d musbat butun sonlar berilganda $(a,b), (a,c), (a,d)$ juftliklarning *EKUB* lari topilsin.
- 6.023** $NOD2$ funksiyasidan foydalanib qisqarmaydigan $\frac{p}{q}$ ko'rinishdagi oddiy kasrni hosil qiluvchi $Frac1(a,b,p,q)$ nomli void turidagi funksiya tasvirlansin. (funksiya barcha parametrlari butun turga tegishli, a va b -kiruvchi, p va q chiquvchi). Bu funksiya dan foydalanib a, b, c, d, e, f, g, h butun sonlar berilganda $\frac{a}{b}+\frac{c}{d}, \frac{a}{b}+\frac{e}{f}, \frac{a}{b}+\frac{g}{h}$ ifodalar uchun qisqarmaydigan kasrlar topilsin.
- 6.024** a va b sonlarining eng kichik umumiy karralisini $a \cdot \frac{b}{NOD(a,b)}$ formula asosida hisoblaydigan butun turga tegishli $NOK2(a,b)$ nomli funksiya tasvirlansin. a,b,c,d butun sonlar berilganda bu funksiya dan foydalanib $(a,b), (a,c), (a,d)$ juftliklar uchun *EKUK*lar topilsin.
- 6.025** $NOD2$ funksiyasidan foydalanib berilgan 3 ta butun musbat sonning *EKUB*ini topuvchi butun turga tegishli $NOD3(a,b,c)$ nomli funksiya tasvirlansin. Bu funksiya dan foydalanib a,b,c,d butun musbat sonlari berilganda $(a,b,c), (a,c,d), (b,c,d)$ uchliklar uchun *EKUB* lar hisoblansin.
- 6.026** Berilgan t sekundni (h,m,s) soat, minut, sekund formatiga o'tkazuvchi (barcha parametrlar butun turga tegishli, t -kiruvchi, h, m va s -chiquvchi parametrlar) $TimeToHMS(t,h,m,s)$ nomli void turidagi funksiya tasvirlansin. Bu funksiya dan foydalanib t ning berilgan 3 ta har xil qiymatlari uchun soat, minut, sekundlar hisoblansin.
- 6.027** Berilgan (h,m,s) soat, minut, sekund formatidagi vaqtni " t " sekundga oshiradigan (h, m, s - kiruvchi va chiquvchi, t kiruvchi parametr, barcha parametrlar butun musbat sonlar) $IncTime(h,m,s,t)$ nomli void turidagi funksiya tasvirlansin. Bu funksiya dan foydalanib berilgan h, m, s formatidagi vaqt t sekundga oshirilsin va hosil bo'lgan natija chiqarilsin.
- 6.028** y berilgan yil kabisa bo'lsa 1 (rost) aks holda 0 (yolg'on) qiymat qaytaruvchi mantiqiy turga tegishli $IsleapYear(y)$ nomli funksiya tasvirlansin. Bu funksiya dan foydalanib berilgan 2 ta

turli butun musbat qiymatlarda funksiyaning qaytaradigan natijalar chop etilsin.

- 6.029** Berilgan yilning m -oyi uchun *IsLeapYear* funksiyasidan foydalanib kunlar sonini hisoblovchi butun turga tegishli *MonthDays*(m,y) nomli funksiya tasvirlansin. Bu funksiyadan foydalanib berilgan 2 ta turli butun musbat qiymatlar uchun funksiya qaytaradigan natijalar chop etilsin.
- 6.030** *MonthDays*(m,y) funksiyasidan foydalanib berilgan, to'g'ri (d,m,y) kun, oy, yil formatdagi sanadan oldingi kun sanasini aniqlovchi *PrevDate*(d,m,y) (d,m,y lar butun turga tegishli kiruvchi va chiquvchi parametrlar) nomli void turidagi funksiya tasvirlansin. Funksiyani qo'llab berilgan 3 ta sananing har biridan oldingi sanalar topilsin.
- 6.031** *MonthDays*(m,y) funksiyasidan foydalanib berilgan, to'g'ri (d,m,y) kun, oy, yil formatdagi sanadan keyingi kun sanasini aniqlovchi *NextDate*(d,m,y) (d,m,y lar butun turga tegishli kiruvchi va chiquvchi parametrlar) nomli void turidagi funksiya tasvirlansin. Funksiyani qo'llab, berilgan 3 ta sananing har biridan keyingi sanalar topilsin.
- 6.032** Uchlarining koordinatalari berilgan kesma uzunligini hisoblovchi haqiqiy turga tegishli *Leng*(x_a, y_a, x_b, y_b) (x_a, y_a, x_b, y_b lar haqiqiy turga tegishli parametrlar) nomli funksiya tasvirlansin. Bu funksiyadan foydalanib, tekislikdagi A, B, C, D nuqtalar koordinatalari berilganda AB, AC, AD kesmalar uzunliklari hisoblansin.
- 6.033** *Leng* funksiyasidan foydalanib uchburchak uchlarining koordinatalari berilganda, uning perimetrini hisoblaydigan haqiqiy turga tegishli *Perim*($x_a, y_a, x_b, y_b, x_c, y_c$) ($x_a, y_a, x_b, y_b, x_c, y_c$ lar haqiqiy turga tegishli parametrlar) nomli funksiya tasvirlansin. Bu funksiyadan foydalanib, tekislikdagi A, B, C, D nuqtalarning koordinatalari berilganda ABC, ABD, ACD uchburchaklarning perimetrlari hisoblansin.
- 6.034** *Leng* va *Perim* funksiyalaridan foydalanib uchburchak uchlarining koordinatalari berilganda uning yuzasini hisoblovchi haqiqiy turga tegishli *Area*($x_a, y_a, x_b, y_b, x_c, y_c$) ($x_a, y_a, x_b, y_b, x_c, y_c$ lar haqiqiy turga tegishli parametrlar) nomli funksiya tasvirlansin. Bu funksiyadan foydalanib, tekislikdagi A, B, C, D nuqtalarning koordinatalari berilganda ABC, ABD, ACD uchburchaklarning yuzalari hisoblansin.

- 6.035** *Leng* va *Area* funksiyalaridan foydalanib uchlarining koordinatalari berilgan kesmadan unda yotmaydigan nuqtagacha bo'lgan masofani hisoblaydigan haqiqiy turga tegishli $Disp(x_p, y_p, x_a, y_a, x_b, y_b)$ ($x_p, y_p, x_a, y_a, x_b, y_b$ lar haqiqiy turga tegishli parametrlar) nomli funksiya tasvirlansin. x_p, y_p nuqta koordinatalari, x_a, y_a, x_b, y_b kesma oxirlarining koordinatalari. Bu funksiya foydalanib, tekislikdagi P, A, B, C, D nuqtalar berilganda P nuqtadan AB, CD, BC kesmalargacha bo'lgan masofalar topilsin.
- 6.036** *Dist* funksiyasidan foydalanib uchlarining koordinatalari berilgan uchburchakning tomonlariga tushirilgan balandliklarini hisoblovchi $Heights(x_a, y_a, x_b, y_b, x_c, y_c, h_a, h_b, h_c)$ ($x_a, y_a, x_b, y_b, x_c, y_c, h_a, h_b, h_c$ lar haqiqiy turga tegishli parametrlar) nomli void turidagi funksiya tasvirlansin. Bu funksiya yordamida tekislikdagi A, B, C, D nuqtalar koordinatalari bilan berilgan bo'lsa ABC, ABD, ACD uchburchaklarning barcha balandliklari hisoblansin.

6.2. Funksiyalarda ko'rsatkichlar va murojaatlar

Ko'pgina masalalarni yechishda funksiyaning formal parametrlaridan qiymat qaytaruvchi sifatida foydalanishga to'g'ri keladi, ya'ni funksiyaning formal parametri bir vaqtning o'zida qiymat qabul qiluvchi va qaytaruvchi bo'lishi talab etiladi. Buning uchun ko'rsatkichlar va murojaatlardan foydalaniladi.

Faktik parametr adres bilan uzatilganda unga mos keluvchi formal parametrni ikki xil usul bilan yozish mumkin: ko'rsatkich yoki murojaat orqali. Ko'rsatkich orqali yozilganda formal parametr turidan keyin '*' belgisi yoziladi, mos argumentda esa o'zgaruvchining adresi (& amal orqali) yoki massiv nomi, yoki funksiya nomi bo'lishi mumkin. Murojaat orqali parametr uzatishda formal parametrda turidan keyin '&' belgisi yoziladi va funksiya chaqirilishida mos argument sifatida o'zgaruvchi nomi keladi.

Misol:

```
#include <iostream.h>
void f(int,int*,int&)
void main()
{
  int i=1, j=2, k=3;
```

```

cout<<i<<" "<<j<<" "<<k;
f(i,&j,k);
cout<<i<<" "<<j<<" "<<k;
}
void f(int i; int*j; int &k)
{
i++;
(*j)++;
k++;
*j=i+k;
k=*j+i;
}

```

Dastur ishlashi natijasida ekranga quyidagi qiymatlar chop etiladi:

1 2 3

1 6 8

Bu misolda birinchi parametr *i*, bir qiymati bilan uzatiladi ("*int i*"). Uning qiymati funksiya ichida o'zgaradi, lekin asosiy funksiyadagi *i* ning qiymati o'zgarmaydi. Ikkinchi parametr ko'rsatkich orqali adresi bilan uzatilishi talab qilinadi ("*int j*"), adresni uzatish uchun '&'-adresni olish amali ishlatilgan ("&j"). Funksiya tanasida argument adresidan qiymat olish uchun '*'-qiymat olish amali ishlatilgan. Uchinchi parametrda murojaat orqali ("&k") argumentning adresini uzatish ko'zda tutilgan. Bu holda funksiya chaqirilishida mos argument o'rnida o'zgaruvchi nomi turadi, funksiya ichida esa qiymat olish amalini ishlatishning hojati yo'q. Funksiya ishlashi natijasidagi qiymatlarni argumentlar ro'yxati orqali olish qulay va tushunarli usul hisoblanadi.

Agar funksiya ichida adres bilan uzatiladigan parametrning qiymati o'zgarmasdan qolishi zarur bo'lsa, bu parametr *const* modifikator bilan yozilishi kerak.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

6.037 *a* sonining 3-darajasini hisoblab, *b* o'zgaruvchiga uzatadigan *PowerA3(a,b)* nomli void turidagi funksiya tasvirlansin (*a*-kiruvchi, *b*-chiquvchi parametrlar bo'lib, ular haqiqiy turga tegishli). Bu funksiya yordamida berilgan 5 ta sonning 3-darajalari hisoblansin.

6.038 *a* sonining 2-, 3-, 4-darajalarini hisoblab, mos ravishda *b,c,d*

- parametrlarga uzatadigan *PowerA234(a,b,c,d)* nomli void turidagi funksiya tasvirlansin (*a*-kiruvchi, *b,c,d* lar chiquvchi parametrlar, barcha parametrlar haqiqiy turga tegishli). Bu funksiya yordamida berilgan 5 ta sonning har biri uchun 2-, 3-, 4-darajalar hisoblansin.
- 6.039** ta *x* va *y* musbat sonlarning o'rtta arifmetigi va o'rtta geometrigini hisoblaydigan *Mean(x,y, Amean, Gmean)* nomli void turidagi funksiya tasvirlansin(*x* va *y* – kiruvchi, *Amean* va *Gmean*-chiquvchi parametrlar bo'lib, ular haqiqiy turga tegishli). Bu funksiyadan foydalanib *a, b, c, d* sonlari berilganda (*a,b*), (*a,c*), (*a,d*) juftliklarning o'rtta arifmetiklari va o'rtta geometriklari hisoblansin.
- 6.040** Tomoni *a* ga teng bo'lgan teng tomonli uchburchakning perimetri va yuzasini hisoblovchi *TrianglePS(a,p,s)* nomli void turidagi funksiya tasvirlansin(*a*-kiruvchi, *p* va *s* chiquvchi parametrlar, barcha parametrlar haqiqiy turga tegishli). Bu funksiyadan foydalanib berilgan *a₁, a₂, a₃* qiymatlarda 3 ta teng tomonli uchburchaklarning perimetrlari va yuzalari hisoblansin.
- 6.041** Tomonlari koordinata o'qlariga parallel bo'lgan hamda qarama-qarshi uchlarining (*x₁, y₁*), (*x₂, y₂*) koordinatalari berilgan to'g'ri to'rtburchakning *p* perimetri va *s* yuzasini hisoblaydigan *RectPS(x1,y1,x2,y2,p,s)* nomli void turidagi funksiya tasvirlansin. Bu funksiyadan foydalanib, qarama-qarshi uchlarining koordinatalari ma'lum, tomonlari koordinata o'qlariga parallel bo'lgan 3 ta to'g'ri to'rtburchakning perimetrlari va yuzalari hisoblansin.
- 6.042** Butun musbat *k* sonidagi raqamlar miqdorini va yig'indisini hisoblaydigan *Digitcountsum(k,n,s)* nomli void turidagi funksiya tasvirlansin (*k*-kiruvchi, *n* va *s* – chiquvchi parametrlar bo'lib, ular butun turga tegishli). Bu funksiyadan foydalanib berilgan 5 ta butun musbat sonning har biri uchun raqamlari miqdori va raqamlar yig'indisi hisoblansin.
- 6.043** Berilgan *k* butun musbat sonining raqamlarini teskari tartibda joylashtiruvchi (123 -> 321) *InvertDigits(k)* nomli void turidagi funksiya tasvirlansin(*k* butun turga tegishli parametr bo'lib, bir vaqtda kiruvchi va chiquvchi hisoblanadi). Bu funksiyadan foydalanib, berilgan 5 ta butun musbat sonlarning raqamlari teskari tartibda joylashtirilsin.

- 6.044** Berilgan butun musbat k sonining o'ng tomonidan " d " raqamni joylashtiradigan $ADDRightDigit(d,k)$ nomli void turidagi funksiya tasvirlansin(d - kiruvchi parametr, k - bir vaqtda kiruvchi va chiquvchi parametrlar bo'lib, ular butun turga tegishli). Bu funksiya dan foydalanib, berilgan n musbat sonining oxiriga d_1 va d_2 raqamlari ketma-ket joylashtirilsin va har bir jarayon natijasi chop etilsin.
- 6.045** Berilgan butun musbat k sonining chap tomonidan " d " raqamini joylashtiradigan $ADDLeftDigit(d,k)$ nomli void turidagi funksiya tasvirlansin(d - kiruvchi parametr, k - bir vaqtda kiruvchi va chiquvchi parametrlar bo'lib, ular butun turga tegishli). Bu funksiya dan foydalanib berilgan k musbat sonining oldidan d_1 va d_2 raqamlari ketma-ket joylashtirilsin va har bir jarayon natijasi chop etilsin.
- 6.046** x va y o'zgaruvchilaridagi saqlanayotgan qiymatlarni o'zaro almashtiruvchi $Swap(x,y)$ nomli void turidagi funksiya tasvirlansin(x, y -haqiqiy turga tegishli parametrlar bo'lib, bir vaqtning o'zida kiruvchi va chiquvchi hisoblanadi). Bu funksiya dan foydalanib berilgan a, b, c, d haqiqiy turga tegishli o'zgaruvchilar uchun $(a,b),(c,d)$ va (b,c) juftliklarning qiymatlari ketma-ket almashtirilsin hamda a, b, c, d larning yangi qiymatlari chiqarilsin.
- 6.047** Berilgan x va y o'zgaruvchilardagi qiymatlarning kichigini x ga kattasini y ga yozadigan $Minmax(x,y)$ nomli void turidagi funksiya tasvirlansin(x,y - haqiqiy turga tegishli parametrlar bo'lib bir vaqtning o'zida kiruvchi va chiquvchi). Bu funksiya 4 marta chaqirilib, berilgan a,b,c, d sonlarining eng kattasi va eng kichigi aniqlansin.
- 6.048** Berilgan a, b, c o'zgaruvchilarning qiymatlarini o'sish tartibida joylashtiruvchi $SortInc3(a, b, c)$ nomli void turidagi funksiya tasvirlansin(a, b, c lar – haqiqiy turga tegishli parametrlar bo'lib, bir vaqtning o'zida kiruvchi va chiquvchi hisoblanadi). Bu funksiya dan foydalanib, berilgan 2 ta (a_1, b_1, c_1) va (a_2, b_2, c_2) sonlar ketma ketliklarining har biri o'sish tartibida tartiblansin.
- 6.049** Berilgan a, b, c o'zgaruvchilarning qiymatlarini kamayish tartibida joylashtiruvchi $SortDec3(a,b,c)$ nomli void turidagi funksiya tasvirlansin(a, b, c lar –haqiqiy turga tegishli parametrlar bo'lib, bir vaqtning o'zida kiruvchi va chiquvchi hisoblanadi). Bu funksiya dan foydalanib berilgan 2 ta (a_1,b_1 ,c_1)

va (a_2, b_2, c_2) sonlar ketma ketliklarining har biri kamayish tartibida tartiblansin.

- 6.050** Berilgan a, b, c o'zgaruvchilardagi qiymatlarni quyidagicha siljituvchi $a \rightarrow b, b \rightarrow c, c \rightarrow a$ o'ng siljish *Shift Right3(a,b,c)* nomli void turidagi funksiya tasvirlansin (a, b, c lar haqiqiy turga tegishli parametrlar bo'lib, bir vaqtning o'zida kiruvchi va chiquvchi hisoblanadi). Bu funksiya foydalanib berilgan 2 ta (a_1, b_1, c_1) va (a_2, b_2, c_2) sonlar ketma ketligining har biri uchun o'ng siljish amali bajarilsin.
- 6.051** Berilgan a, b, c o'zgaruvchilardagi qiymatlarni quyidagicha siljituvchi $a \rightarrow c, c \rightarrow b, b \rightarrow a$ chap siljish *Shift Left3(a,b,c)* nomli void turidagi funksiya tasvirlansin (a, b, c lar haqiqiy turga tegishli parametrlar bo'lib, bir vaqtning o'zida kiruvchi va chiquvchi hisoblanadi). Bu funksiya foydalanib, berilgan 2 ta (a_1, b_1, c_1) va (a_2, b_2, c_2) sonlar ketma ketligining har biri uchun chap siljish amali bajarilsin.

6.3. Rekursiya

Rekursiv funksiyalar ikki xil ko'rinishda tashkil etiladi:

Funksiyaning o'z tanasida o'ziga murojaat qilishiga *oddiy rekursiya* deyiladi.

Agar birinchi funksiya ikkinchi funksiya murojaat qilsa, ikkinchisi esa o'z navbatida birinchi funksiya murojaat qilsa, bu turdagi murojaatga *vositali rekursiya* deyiladi.

Odatda rekursiya matematikada keng qo'llaniladi. Chunki aksariyat matematik formulalar rekursiv aniqlanadi.

Masala. n faktorialni hisoblovchi *rekursiv* funksiya tuzilsin.

$$n! = \begin{cases} 1, & \text{agar } n=0; \\ n*(n-1)!, & \text{agar } n>0; \end{cases}$$

Ko'rinib turibdiki, navbatdagi qiymatni hisoblash uchun funksiyaning oldingi qiymati ma'lum bo'lishi kerak. C++ tilida rekursiya matematikadagi rekursiyaga o'xshash. Buni yuqoridagi misol uchun tuzilgan funksiya ko'rish mumkin:

```
long fac(int n)
{
    if(!n) return 1;
    else return n*fac(n-1);
}
```

Agar faktorial funksiyasiga $n > 0$ qiymat berilsa, quyidagi holat ro'y beradi: shart operatorining *else* tarmog'idagi qiymati (n ning qiymati) stekda eslab qolinadi. Noma'lumlarni hisoblash uchun shu funksiyaning o'zi oldingi qiymat ($n-1$ ning qiymati) bilan chaqiriladi. O'z navbatida, bu qiymat ham eslab qolinadi (stekka joylanadi) va yana funksiya chaqiriladi va hokazo. Funksiya $n=0$ qiymat bilan chaqirilganda *if* operatorining sharti *rost* bo'ladi va «*return 1;*» amali bajarilib, ayni shu chaqirish bo'yicha 1 qiymati qaytariladi. Shundan keyin «teskari» jarayon boshlanadi – stekda saqlangan qiymatlar ketma-ket olinadi va ko'paytiriladi: oxirgi qiymat aniqlangandan keyin (1) , u o'zidan oldingi stekda saqlangan qiymatga, ya'ni 1 qiymatiga ko'paytirib $fac(1)$ qiymati hisoblanadi, bu qiymat 2 qiymatiga ko'paytirish bilan $fac(2)$ hisoblanadi va hokazo.

Rekursiv funksiyalarning to'g'ri ishlashi uchun rekursiv chaqirishlarning to'xtash sharti bo'lishi kerak. Aks holda rekursiya to'xtamasligi va o'z navbatida funksiya ishi cheksiz davom etadi. Faktorialni hisoblashda *rekursiv* tushishlarning to'xtash sharti funksiya parametri $n=0$ bo'lishidir (shart operatorining *rost* tarmog'i).

Har bir *rekursiv* murojaat qo'shimcha xotira talab qiladi– funksiyalarning lokal o'zgaruvchilari uchun har bir murojaatda stekdan yangidan joy ajratiladi. Masalan, rekursiv funksiya 100 marta murojaat qilinsa, jami 100 ta lokal ob'yektlarning majmuasi uchun stekdan joy ajratiladi. Ayrim hollarda, juda ko'p rekursiya bo'lganda, stek o'lchami cheklanganligi sababli u to'lib qolishi mumkin va bu holatda dastur o'z ishini «*Stek to'lib ketdi*» xabari bilan to'xtatadi.

Lekin shunday masalalar borki, ularni yechishda rekursiya juda samarali, hattoki, yagona usuldir. Xususan, grammatik tahlil masalalarida rekursiyadan foydalanish, qulay usullardan biri hisoblanadi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 6.052** $n!$ ni hisoblaydigan haqiqiy turga tegishli $Fact(n)$ funksiyasi tasvirlansin. Berilgan 5 ta son uchun tasvirlangan funksiya foydalanib, faktoriallar hisoblansin.
- 6.053** Ikkilangan faktorialni hisoblovchi ($n!! = n \cdot (n-2) \cdot (n-4) \cdot \dots$) $n > 0$ haqiqiy turga tegishli $Fact2(n)$ funksiyasi tasvirlansin. Berilgan 5 ta son uchun bu funksiya yordamida ikkilangan faktoriallar

hisoblansin.

6.054 x ning n -darajasini hisoblaydigan, ($x^0=1$, n juft bo'lganda

$$x^n = \left(x^{\frac{n}{2}}\right)^2 \quad (n>0), \quad n \text{ toq bo'lganda } x^n = x \cdot x^{n-1} \quad n>0, \quad n<0 \text{ bo'lganda}$$

$$x^n = \frac{1}{x^{-n}} \quad \text{haqiqiy turga tegishli } PowerN(x,n) \text{ funksiyasi}$$

tasvirlansin. (x haqiqiy turga tegishli ($x \neq 0$), n -butun turga tegishli). Bu funksiya yordamida n ning 5 ta turli qiymatlaridagi x ning darajalari hisoblansin.

6.055 Fibonachchi sonlarining n -hadini hisoblaydigan (n -butun son) butun turga tegishli $Fib1(n)$ funksiyasi tasvirlansin. Bu funksiya yordamida turli 5 ta butun son uchun Fibonachchi sonlari hisoblansin, har bir n uchun $Fib1$ funksiyasini chaqirilishlar soni ham aniqlansin.

6.056 n ta elementli Fibonachchi sonlari ketma-ketligini hisoblaydigan butun turga tegishli $Fib2$ funksiyasi tasvirlangan. Bunda $n \leq 20$. yordamchi massiv tuzilsin va unga hisoblangan fibonachi sonlari yozilib, $Fib2$ funksiyasi bajarilayotganda murojaat etilsin. $Fib2$ funksiyasi yordamida n ning 5 ta turli har xil qiymatlari uchun Fibonachchi sonlari hisoblansin.

6.057 $c(n,k)$ ($c(n,0)=c(n,n)=1$, $c(n,k)=c(n-1,k)+c(n-1, k-1)$ $0 < k < n$ bo'lsa) ifodani hisoblovchi butun turga tegishli $Combin1(n,k)$ funksiyasi tasvirlansin. Bu funksiya yordamida $n > 0$, $0 \leq k \leq n$ bo'lgan holda, berilgan n soni uchun k ning turli 3 ta qiymatida $c(n,k)$ soni hisoblansin va $c(n,k)$ ni hisoblashda $Combin1$ funksiyani chaqiruvlar soni ham aniqlansin.

6.058 $c(n,k)$ ($c(n,0)=c(n,n)=1$, $c(n,k)=c(n-1,k)+c(n-1, k-1)$ $0 < k < n$ bo'lsa) ifodani hisoblaydigan butun turga tegishli $Combin2(n,k)$ funksiya tasvirlansin (n, k parametrlari butun turga tegishli). Bunda $n \leq 20$. $c(n,k)$ ning hisoblangan qiymatlarini o'zida saqlovchi ikki o'lchovli massiv tuzilsin. $Combin2$ da bu massivdan foydalanilsin. Ushbu funksiya yordamida berilgan n uchun k ning 5 ta turli qiymatlarida $c(n,k)$ hisoblansin, har bir hisoblash jarayonida hosil bo'lgan qiymatlar ikki o'lchovli massivda tasvirlansin.

6.059

$$k \text{ tartibli ildizni hisoblaydigan } (y_0=1, \quad y_{n+1} = y_n - \frac{y_n - \frac{x}{y_n^{k-1}}}{k})$$

haqiqiy turga tegishli $RootK(x, k, n)$ funksiyasi tasvirlansin. Bu yerda y_n $RootK(x, k, n)$ funksiyadagi fikserlangan x va k ni bildiradi. Funksiya parametrlari: $x(x > 0)$ -haqiqiy son $k(k > 1)$ va $n(n > 0)$ -butun sonlar. $RootK$ funksiyasi yordamida berilgan x soni uchun uning k -tartibli ildizini n ning 6 ta turli qiymatlarida hisoblansin.

- 6.060** 2 ta butun sonning umumiy bo'luvchisini Evklid algoritmi yordamida topadigan ($NOD(a, b) = NOD(b, a \% b)$, agar $b \neq 0$; $NOD(a, 0) = a$) butun turga tegishli $NOD(a, b)$ funksiyasi tasvirlansin. Agar a, b, c, d musbat butun sonlar berilgan bo'lsa NOD funksiyasidan foydalanib, (a, b) , (a, c) , (a, d) juftliklar uchun EKUB lar hisoblansin.
- 6.061** k sonining raqamlari yig'indisini hisoblaydigan butun turga tegishli $DigitSum(k)$ funksiyasi tasvirlansin, raqamlar yig'indisini hisoblashda sikl operatorlaridan foydalanilmasin. Bu funksiya yordamida turli 5 ta sonning raqamlari yig'indisini hisoblash amalga oshirilsin.
- 6.062** Sikl operatoridan foydalanmasdan $n(1 \leq n \leq 10)$ ta elementga ega a massivdagi eng katta elementni topuvchi butun turga tegishli $MaxElem(a, n)$ funksiyasi tasvirlansin. Ushbu funksiya yordamida a, b, c masivlarning har biridagi eng katta elementlar topilsin.
- 6.063** s satrdagi berilgan sonning raqamlari miqdorini topadigan butun turga tegishli $DigitCount(s)$ funksiyasi tasvirlansin. Hisoblash jarayonida sikl operatoridan foydalanilmasin. Berilgan 5 ta har xil sonlardan iborat satrlardagi son raqamlari yig'indisi, tasvirlangan funksiyadan foydalanib hisoblansin.
- 6.064** *Polindrom* sonni aniqlaydigan mantiqiy turga tegishli $Polindrom(S)$ funksiyasi tasvirlansin. Agar s son palindrom bo'lsa funksiya 1 (rost), aks holda 0 (*yolg'on*) qiymat qaytarsin. (*Polindrom* son bu-chapdan o'ngga va o'ngdan chapga bir xil o'qiladigan sonidir.) Berilgan 5 ta turli sonlar uchun *Polindrom* funksiyasining qaytaradigan qiymatlari chiqarilsin.

6.4. Bobga doir qiyinroq masalalar

6.065

Havfli zona [12]

Yo'ldosh, Shavkat va G'iyos $n \times m$ lik maydonda (maydon $n \times m$ ta kvadratlardan tashkil topgan) ahil yashashardi. Bir kuni

Yo'ldosh va Shavkat arzimagan narsa ustida tortishib qoldi, shundan so'ng Shavkat maydondan o'ziga tegishli hududni ajratib oldi. Shavkatning hududiga k yoki undan kichik masofagacha yaqin kelish Yo'ldosh uchun juda havfli. Maydondagi (x_1, y_1) katakchadan, (x_2, y_2) katakchagacha bo'lgan masofa $|x_2 - x_1| + |y_2 - y_1|$ ga teng. Yo'ldosh Shavkat bilan kelishib olishni muhokama qilish uchun G'iyosning uyiga borishi kerak. Yo'ldosh bir urinishda o'zi turgan katakdan qo'shni kataklarning biriga yura oladi. Ikki katak umumiy tamonga ega bo'lsa qo'shni deb aytiladi. Maydonning tashqarisiga chiqib bo'lmaydi va Shavkatning hududiga k yoki undan kichik masofaga yaqin bormasligi kerak. Yo'ldoshga bu sharoitda G'iyosning uyiga boradigan eng qisqa yo'lni topishda yordam bering.

Berilganlar:

Birinchi qatorda n , m va k sonlari berilgan, n –qatorlar soni, m –ustunlar soni ($1 \leq n, m \leq 500$, $1 \leq k \leq 1000$). Keyingi n ta qatorning har birida m tadan simvol beriladi. Bo'sh kataklar '.'(nuqta), Shavkatga tegishli kataklar 'S'-harfi, Yo'ldosh turgan katak 'Y' harfi, G'iyos turgan katal 'G' harfi bilan beriladi. Y va G harflari bir martadan uchrashi va Yo'ldosh turgan katak havfli emasligi kafolatlanadi.

Natija:

Agar Yo'ldosh Gi'yosning uyiga boraolmasa -1, aks holda unga boradigan eng qisqa yo'l uzunligini chiqarilsin.

<i>N_o</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	3 3 0 YSG ... SSS	4
2	3 5 1 Y.S.G	8
3	3 5 1 Y.SG.	-1

6.066

Uchburchak tort [12]

Yaqinda Shavkatning tug'ilgan kuni bo'ldi. G'iyos unga uchburchak shaklida tort pishirdi, Yo'ldosh esa unga yordamlashdi. Ular doirani juda yoqtirishadi. Shuning uchun ularning har biri tortdan doira shaklida bo'lak kesib olmoqchi. Tug'ilgan kun Shavkatniki bo'lganligi uchun birinchi bo'lib u uchburchak shaklidagi tortdan eng katta yuzali doira shaklidagi qismini kesib oldi. Asosiy oshpaz G'iyos bo'lganligi uchun ikkinchi bo'lib u qolgan qismdan eng katta yuzali doira shaklidagi qismni kesib oldi. Uchinchi bo'lib esa Yo'ldosh qolgan qismlardan yuzasi eng katta bo'ladigan qilib doira shaklida tort bo'lagini kesib oldi. Yeyishni boshlashdan oldin ularni bir savol qiziqtirib qo'ydi: ularning har biri qanchadan tort kesib olgan? Bu savolga javob topishda ularga yordam bering.

Berilganlar:

Birinchi qatorda uchta butun son, uchburchak tort tomonlarining uzunliklari a, b, c lar berilgan ($1 \leq a, b, c \leq 1000$). Bu sonlar uchburchak tengsizligini qanoatlantiradigan qilib beriladi.

Natija:

Shavkat, G'iyos va Yo'ldoshlarning tort bo'laklari yuzalarini 10-6 aniqlikda bittadan bo'sh joy bilan ajratib chiqaring.

<i>Berilganlar:</i>	<i>Natija:</i>
6 7 8	11.780972
	2.212878
	1.423423

6.067

Tosh olish o'yini [10].

Yaqin do'stlar Prince va Starle bo'sh vaqtlarida har xil o'yinlarni o'ynashni yaxshi ko'radi. Shulardan biri *Tosh olish* o'yini va u mana bunday ko'rinishda o'ynaladi: Ularda N ta toshdan iborat bo'lgan toshlar to'plami bor. O'yin qoidasi bo'yicha o'yinchilarning har biri o'z navbati bilan 2 ning manfiy bo'lmagan darajalari (1, 2, 4, 8 va hokazolar) miqdoridagi toshlarni olishi mumkin. O'yinda oxirgi toshni olgan o'yinchi g'olib hisoblanadi va o'yinni har doimgidek Prince boshlab beradi. Ana endi Prince yutishi uchun oldiniga qancha miqdorda minimal tosh olishi kerakligi haqida o'ylab qoldi. Sizning vazifangiz o'yinda kim g'olib bo'lishini aniqlash

va yana agarda Prince g'olib bo'ladigan bo'lsa u birinchi yurishida qancha miqdorda minimal tosh olishi kerakligini topish.

Berilganlar:

Bitta butun son N ($1 \leq N \leq 10250$).

Natija:

O'yinda Prince g'olib bo'lsa "Prince" yozuvini, aks holda "Starle" yozuvini chiqaring. Agar o'yinda Prince go'lib bo'lsa ikkinchi qatorga unga g'olib bo'lishi uchun birinchi yurishda olish kerak bo'ladigan minimal toshlar sonini chiqaring. Shuni yodda tutingki Starle va Prince bu o'yinning ustalari hisoblanadi. Shuning uchun ular o'zlarining har bir yurishida o'zlari uchun eng yaxshi strategiya (optimal yurish)ni tanlaydilar.

<i>No</i>	<i>Berilganlar</i>	<i>Natija</i>
1	8	Prince 2
2	6	Starle
3	5	Prince 2

6.068

Kataklarni bo'yash o'yini [10]

Yaqin do'stlar Prince va Starle bo'sh vaqtlarida har xil o'yinlarni o'ynashni yaxshi ko'radi. Shulardan biri *Kataklarni bo'yash* o'yini va u mana bunday ko'rinishda o'ynaladi: Ularda N ta ketma-ket joylashgan katakchalardan iborat doska bor. Har bir o'yinchi o'z navbati bilan doskaning ixtiyoriy joydagi hali bo'yalmagan K ta ketma-ket katakni bo'yaydi. O'yinchi o'yinda o'z yurishini bajara olmasa (yani unga K ta ketma-ket bo'yalmagan katak qolmagan bo'lsa) u yutqazgan hisoblanadi. O'yinni har doimgidek Prince boshlab beradi. Sizning vazifangiz o'yinda kim g'olib bo'lishini aniqlashdan iborat.

Berilganlar:

Bitta qatorda ikkita butun sonlar N ($1 \leq N \leq 100$) va K ($1 \leq K \leq N$) beriladi.

Natija:

O'yinda Prince g'olib bo'lsa "*Prince wins*" yozuvini, aks holda "*Starle wins*" yozuvini chiqaring. Shuni yodda tutingki, Starle va Prince bu o'yinning ustalari hisoblanadi. Shuning uchun ular o'zlarining har bir yurishida o'zlari uchun eng yaxshi strategiya (optimal yurish)ni tanlaydi.

<i>N_o</i>	<i>Berilganlar</i>	<i>Natija</i>
<i>1</i>	<i>5 2</i>	<i>Starle wins</i>
<i>2</i>	<i>5 3</i>	<i>Prince wins</i>
<i>3</i>	<i>5 1</i>	<i>Prince wins</i>

VII Bob. Fayllar bilan ishlash

Algoritmik va ob'ektga mo'ljallangan dasturlar tillaridagi standart va dasturchi tomonidan aniqlangan turlarning muhim xususiyatlaridan biri, ularning oldindan berilgan chekli elementlardan tashkil topganligidadir. Berilganlar dinamik aniqlanganda ham, operativ xotiraning cheklanganligi sababli, berilganlar miqdori yuqoridan chegaralangan elementlardan iborat bo'ladi. Bir qator tadbqiqiy masalalar uchun oldindan berilganning komponentalari sonini aniqlash imkoni yo'q. Ular masalani yechish jarayonida aniqlanadi hamda yetarlicha katta hajmda bo'lishi mumkin. Qolaversa, dasturda e'lon qilingan o'zgaruvchilarning qiymatlari sifatida aniqlangan berilganlar faqat dastur ishlash paytidagina mavjud bo'ladi va dastur o'z ishini tugatgandan keyin yo'qolib ketadi. Agar dastur yangidan ishga tushirilsa, berilganlarni yangidan hosil qilish zarur bo'ladi. Aksariyat tadbqiqiy masalalar esa berilganlarni doimiy ravishda saqlab turishni talab qiladi. Bu talabga javob berish uchun fayl turidagi ob'yektlar (o'zgaruvchilar) qo'llaniladi.

Fayl – bu bir xil turdagi qiymatlar joylashgan tashqi xotiradagi nomlangan sohadir.

Faylni, boshida ketma-ket ravishda joylashgan yozuvlar bilan to'ldirilgan va oxiri bo'sh bo'lgan yetarlicha uzun magnit tasmasiga o'xshatish mumkin.

Yana bir muhim tushunchalardan biri fayl ko'rsatkichi tushunchasidir. Fayl ko'rsatkichi – ayni paytda fayldan o'qilayotgan yoki unga yozilayotgan joyni ko'rsatib turadi, ya'ni fayl ko'rsatkichi ko'rsatib turgan joydan yozuvni o'qish yoki shu joyga yangi yozuvni joylashtirish mumkin.

Fayldagi yozuvlarga murojaat ketma-ket ravishda amalga oshiriladi. Shuni ta'kidlab o'tish zarurki, fayldan yozuvlarni o'qish jarayoni qisman avtomatlashtirilgan, ya'ni *i*-yozuv o'qilgandan keyin, fayl ko'rsatkichi navbatdagi $(i+1)$ -yozuv boshiga o'tadi va shu tarzda o'qish davom ettiriladi. *Fayl*—bu berilganlarni saqlash joyidir va shu sababli uning yozuvlari ustida to'g'ridan-to'g'ri amal bajarib bo'lmaydi. Fayl yozuvi ustida amal bajarish uchun yozuv qiymati operativ xotiraga mos turdagi o'zgaruvchiga o'qilishi kerak. Keyinchalik, zarur amallar shu o'zgaruvchi ustida bajariladi, kerak bo'lsa natijalar yana faylga yozilishi mumkin.

Operasion sistema nuqtai-nazaridan fayl hisoblangan ixtiyoriy ma'lumot C++ tili uchun moddiy fayl hisoblanadi. Fayl nomlari satr

turidagi o'zgarmlar yoki o'zgaruvchilarga qiymatlanadi. Operatsion sistema qoidalariga ko'ra fayl nomi to'liq bo'lishi, ya'ni fayl nomining boshida adres qismi keltirilishi shart: Masalan,

C:\BORLAND\BIN\Masala.cpp.

C++ tilida mantiqiy fayl tushunchasi kiritilgan bo'lib, u fayl turidagi o'zgaruvchidir. Fayl turidagi o'zgaruvchilarga boshqa turdagi o'zgaruvchilar kabi qiymat berish operatori orqali qiymat berib bo'lmaydi. Boshqacha aytganda fayl turidagi o'zgaruvchilar ustida hech qanday amal aniqlanmagan. Ular ustida bajariladigan barcha amallar funksiyalar vositasida amalga oshiriladi.

Fayllar bilan ishlash quyidagi bosqichlardan iborat:

- *fayl o'zgaruvchisi diskdagi fayl bilan bog'lanadi;*
- *fayl ochiladi;*
- *fayl ustida yozish yoki o'qish amallari bajariladi;*
- *fayl yopiladi.*

7.1. Sonli va satrli fayllar

Fayllar *binar (ikkilik)* yoki *matn* ko'rinishida bo'lishi mumkin.

Binar fayllar—bu oddiygina baytlar ketma-ketligidir. *Binar* fayllardan berilganlarni foydalanuvchi tomonidan bevosita ko'rish zarur bo'lmagan hollarda foydalaniladi. Binar fayllaridan o'qish-yozishda baytlar ustida hech qanday konvertatsiya amallari bajarilmaydi.

Endi fayllar bilan ishlashda kerak bo'ladigan bir qator tushunchalar bilan tanishamiz. Oqim tushunchasi—bu berilganlarni faylga o'qish-yozishda ularni belgilar ketma-ketligi yoki oqimi ko'rinishida tasavvur qilishdan kelib chiqqan. Oqim ustida quyidagi amallarni bajarish mumkin:

- *oqimdan berilganlar blokini operativ xotiraga o'qish;*
- *operativ xotiradagi berilganlar blokini oqimga yozish (chiqarish);*
- *oqimdagi berilganlar blokini yangilash;*
- *oqimdan yozuvni o'qish;*
- *oqimga yozuvni chiqarish.*

Oqim bilan ishlaydigan barcha funksiyalar buferli, formatlashgan yoki formatlashmagan o'qish-yozishni ta'minlaydi.

Dastur ishga tushganda o'qish-yozishning quyidagi standart oqimlari ochiladi:

stdin – o'qishning standart vositasi;

stdout - yozishning standart vositasi;

stderr - xatolik haqida xabar berishning standart vositasi;

stdprn - qog'ozga chop qilishning standart vositasi;

stdaux - standart yordamchi qurilma.

Kelishuv bo'yicha *stdin*-foydalanuvchi klaviyaturasi, *stdout* va *stderr* - terminal (monitor), *stdprn* - printer bilan, hamda *stdaux* - kompyuter yordamchi portlariga bog'lanish hisoblanadi. Berilganlarni o'qish-yozishda *stderr* va *stdaux* oqimidan boshqa oqimlar buferlanadi, ya'ni belgilar ketma-ketligi operativ xotiraning bufer deb nomlanuvchi sohasida vaqtincha jamlanadi.

Zamonaviy operatsion sistemalarda klaviyatura va displeylar matn fayllari sifatida qaraladi. Haqiqatdan ham berilganlarni klaviyaturadan dasturga kiritish (o'qish) mumkin, ekranga esa chiqarish (yozish) mumkin. Dastur ishga tushganda standart o'qish va yozish oqimlari o'rniga matn fayllarni tayinlash orqali bu oqimlarni qayta aniqlash mumkin. Bu holat o'qish va yozishda qayta adreslash ro'y berdi deyiladi. O'qish uchun qayta adreslashda '<' belgisidan, yozish uchun esa '>' belgisidan foydalaniladi. Misol uchun *gauss.exe* bajariluvchi dastur berilganlarni o'qishni klaviyaturadan emas, balki *massiv.txt* faylidan amalga oshirish zarur bo'lsa, u buyruq satrida quyidagi ko'rinishda yuklanishi zarur bo'ladi:

gauss.exe < massiv.txt

Agar dastur natijasini *natija.txt* fayliga chiqarish zarur bo'lsa,

gauss.exe > natija.txt satri yoziladi.

Va nihoyat, agar berilganlarni *massiv.txt* faylidan o'qish va natijani *natija.txt* fayliga yozish uchun

gauss.exe < massiv.txt > natija.txt buyruq satri teriladi.

Binar va *matnli* fayllar bilan ishlovchi funksiyalar:

Fayl oqimini ochish funksiyasining prototipi quyidagi

*FILE * fopen(const char * filenomi, const char * fotartib);*

ko'rinishida aniqlangan bo'lib, u *fopen()* funksiyasi orqali amalga oshiriladi. Funksiya «*filenomi*» o'zgaruvchisida ko'rsatilgan nomdagi faylni ochadi, ya'ni u bilan oqimni bog'laydi va oqimni identifikasiya qiluvchi ko'rsatkichni javob tariqasida qaytaradi. Faylni ochish muvoffaqiyatsiz bo'lsa, *fopen()* funksiyasi NULL qiymatini qaytaradi.

Parametrlar ro'yxatidagi ikkinchi – «*fotartib*» parametri faylni ochish rejimini aniqlaydi. U qabul qilishi mumkin bo'lgan qiymatlar quyidagi jadvalda keltirilgan:

fotartib qiymati	Faylning ochilish holati tavsifi
R	Faylni, faqat undan ma'lumotni o'qish uchun ochish.
W	Faylni, faqat unga ma'lumot yozish uchun ochish. Agar bunday nomdagi fayl mavjud bo'lsa, undagi ma'lumotlar to'liq yangilanadi.
A	Faylga yozuvni qo'shish rejimi. Agar bunday nomdagi fayl mavjud bo'lsa, fayl o'zining ma'lumotlari oxiriga yangi ma'lumotni qo'shish uchun ochiladi, aks holda yangi fayl yaratilib, u yozish rejimida ochiladi.
R+	Mavjud fayl (o'qish va yozish) o'zgartirish uchun ochiladi.
W+	Yangi fayl yaratilib, (o'qish va yozish) o'zgartirish uchun ochiladi. Agar bunday nomdagi fayl mavjud bo'lsa, undagi oldindan mavjud ma'lumotlar o'chiriladi va u yangi ma'lumotlar yozishga tayyorlanadi.
a+	Faylga yozuvni qo'shish rejimi. Agar berilgan nomdagi fayl mavjud bo'lsa, uning oxiriga (<i>EOF</i> alomatidan keyin) ma'lumotni yozish (o'qish) uchun ochiladi, aks holda yangi fayl yaratilib, u yozish rejimida ochiladi.

7.1-jadval. Fayl ochish rejimlari.

Binar fayl ochilayotganligini bildirish uchun fayl ochilish rejimi satriga 'b' belgisini qo'shib yozish zarur bo'ladi. Misol uchun fayl ochilishining "wb+" rejimi binar fayl yangilanishini bildiradi.

Fayl o'zgartirish (*o'qish-yozish*) uchun ochilganda, berilganlarni oqimdan o'qish, hamda oqimga yozish mumkin. Biroq yozish amalidan keyin darhol o'qib bo'lmaydi, buning uchun o'qish amalidan oldin *fseek()* yoki *rewind()* funksiyalari chaqirilishi shart. Bu funksiyalar tavsifiga keyinroq to'xtalamiz.

Shuni ta'kidlash lozimki, har bir ochilgan fayl u bilan ishlash tugagandan keyin, albatta yopilishi shart. Buning uchun *fclose()* funksiyasidan foydalaniladi. Funksiya prototipi quyidagi ko'rinishga ega:

```
int fclose(FILE * stream);
```

fclose() funksiyasi oqim bilan bog'lik buferlarni tozalaydi (masalan, faylga yozish ko'rsatmasi berilgan, lekin shu paytgacha buferda saqlanayotgan berilganlarni diskdagi faylga yozadi) va faylni yopadi.

Agar faylni yopishda xatolik ro'y bersa, funksiya *EOF* qiymatini, yopish muvaffaqiyatli yakunlanganda esa *0* qiymatini qaytaradi.

Fayl oxirini aniqlash funksiyasi *feof()* bo'lib, uning prototipi quyidagi,

```
int feof(FILE *stream);
```

ko'rinishda aniqlangan. Bu funksiya fayl oxiri belgisi uchrasa, noldan farqli sonni, boshqa holatlarda esa 0 qiymatini qaytaradi.

Fayldan belgini o'qish funksiyasi *fgetc()* bo'lib, uning prototipi quyidagi,

```
int fgetc(FILE *stream);
```

ko'rinishda aniqlangan. Agar o'qish muvaffaqiyatli bo'lsa, funksiya o'qilgan belgini *int* turidagi ishorasiz butun songa aylantiradi. Agar fayl oxirini o'qishga harakat qilinsa yoki xatolik ro'y bersa, bu funksiya *EOF* qiymatini qaytaradi.

Faylga belgini yozish funksiyasi *fputc()* bo'lib, uning prototipi quyidagi,

```
int fputc(int c, FILE *stream);
```

ko'rinishda aniqlangan. *fgetc()* funksiyasi fayl oqimiga argumentda ko'rsatilgan belgini yozadi.

Fayldan satrlarni o'qish funksiyasi *fgetc()* bo'lib, uning prototipi quyidagi,

```
char * fgetc(char * s, int n, FILE *stream);
```

ko'rinishda aniqlangan. Bu funksiya fayl oqimidan belgilar ketma-ketligini *s* satriga o'qiydi. Funksiya o'qish jarayonini oqimdan *n-1* ta belgi o'qilgandan keyin, ya'ni '\n' uchraganda to'xtatadi. Belgilarni o'qish tugagandan keyin *s* satr oxiriga, satr tugash alomati '\0' belgisi qo'shiladi. Agar satrni o'qish muvaffaqiyatli bo'lsa, funksiya *s* argument ko'rsatadigan satrni qaytaradi, aks holda NULL ni qaytaradi.

Fayl oqimiga satrni *fputs()* funksiyasi yordamida chiqarish mumkin. Bu funksiya prototipi quyidagi

```
int fputs(const char *s, FILE *stream);
```

ko'rinishda aniqlangan. Satr oxiridagi yangi satrga o'tish belgisi va terminatorlar oqimga chiqarilmaydi. Oqimga chiqarish muvaffaqiyatli bo'lsa, funksiya nomanfiy son qaytaradi, aks holda *EOF* ni qaytaradi.

Dasturda berilganlarni oqimdan o'qish yoki oqimga chiqarishda fayldan formatli o'qish – *fscanf()* va yozish – *fprintf()* funksiyalaridan foydalaniladi. Bu funksiyalarning mos ravishda *scanf()* va *printf()* funksiyalaridan farqi – ular berilganlarni birinchi argument sifatida beriladigan fayldan o'qiydi va yozadi.

Fayl oqimidan formatlashmagan o'qish-yozishni amalga oshirish funksiyalari:

fread() funksiyasining prototipga quyidagi

*size_t fread(void*ptr, size_t size, size_t n, FILE *stream);*

ko'rinishda aniqlangan bo'lib, bu funksiya oqimdan *ptr* ko'rsatib turgan buferga, har biri *size* bayt bo'lgan *n* ta berilganlar blokini o'qiydi. O'qish muvaffaqiyatli bo'lsa, funksiya o'qilgan bloklar sonini qaytaradi. Agar o'qish jarayonida fayl oxiri uchrab qolsa yoki xatolik ro'y bersa, funksiya to'liq o'qilgan bloklar sonini yoki 0 qiymatini qaytaradi.

fwrite() funksiyasi prototipi quyidagi,

*size_t fwrite(const void*ptr, size_t size, size_t n, FILE *stream);*

ko'rinishda aniqlangan bo'lib, bu funksiya *ptr* ko'rsatib turgan buferdan, har biri *size* bayt bo'lgan *n* ta berilganlar blokini oqimga chiqaradi. Yozish muvaffaqiyatli bo'lsa, funksiya yozilgan bloklar sonini qaytaradi. Agar yozish jarayonida xatolik ro'y bersa, funksiya to'liq yozilgan bloklar sonini yoki 0 qiymatini qaytaradi.

Fayl ko'rsatkichini boshqarish funksiyalari – *fseek()*, *ftell()* va *rewind()* funksiyalari fayl ko'rsatkichini o'zgartirish va uning qiymatini olish imkonini beradi.

ftell() funksiyasining prototipi quyidagi,

*long int ftell(FILE *stream);*

ko'rinishda aniqlangan bo'lib, argumentda ko'rsatilgan faylga xizmat qiluvchi fayl ko'rsatkichining qiymatini, agar xatolik ro'y bersa 1L qiymatini qaytaradi.

fseek() funksiyasining prototipi quyidagi,

*int fseek(FILE *stream, long offset, int from);*

ko'rinishda aniqlangan bo'lib, bu funksiya *stream* fayli ko'rsatkichini «*from*» joyiga nisbatan «*offset*» bayt masofaga surishni amalga oshiradi. Matn rejimidagi oqimlar uchun «*offset*» qiymati 0 yoki *ftell()* funksiyasi qaytargan qiymat bo'ladi. «*from*» parametri quyidagi qiymatlardan birini qabul qila oladi:

SEEK_SET (=0) - fayl boshi;

SEEK_CUR (=1) - fayl ko'rsatkichining ayni paytdagi qiymati;

SEEK_END (=2) - fayl oxiri.

Bu funksiya yordamida fayl ko'rsatkichi qiymatining o'zgartirilishi muvaffaqiyatli amalga oshirilgan bo'lsa, u 0 qiymatini, aks holda noldan farqli qiymatni qaytaradi.

rewind() funksiyasining prototipi quyidagi

*void rewind(FILE *stream);*

ko'rinishda aniqlangan bo'lib, u fayl ko'rsatkichini fayl boshiga olib keladi.

Masala. Binar fayldan haqiqiy sonlarni o'qish (agar fayl mavjud bo'lmasa uni hosil qilish va haqiqiy sonlar bilan to'ldirish) va o'qilgan sonlarning o'rta arifmetigini hisoblash, hamda ushbu sonlar orasidan hisoblangan o'rta arifmetikdan kichiklari miqdorini aniqlash dasturi tuzilsin.

Izoh: Faylni yaratish va undagi o'rta arifmetikdan kichik sonlar miqdorini aniqlash, alohida funksiyalar ko'rinishida tasvirlanishi mumkin.

```
#include <iostream.h>
#include <stdio.h>
#include <string.h>
// Yangi fayl yaratish va unga sonlarni yozish funksiyasi;
int Fayl_tuzish()
{
    FILE *f;
    double x;
    // «f» fayl yangidan hosil qilish uchun ochilmoqda;
    if((f=fopen("Haqiqiy.son", "wb+"))==NULL) return 0;
    char *satr=new char[10];
    int n=1;
    do {
        cout<<"Haqiqiy sonni kiriting: ";
        gets(satr);
        if(strlen(satr))
        { x=atof(satr);
          fwrite (&x,sizeof(double),n,f); } }
    while(strlen(satr));
    // kiritilgan satr bo'sh bo'lmasa, takrorlanish davom etadi;
    fclose(f);
    return 1; }
// O'rta arifmetikdan kichik sonlar miqdorini hisoblash funksiyasi;
int Kichiklar_soni()
{
    FILE*f;
    double x;
    f=fopen("Haqiqiy.son", "rb+");
    double s=0; // s - f fayl elementlari yig'indisi;
    while(!feof(f))
    {
```

```
    if (fread(&x,sizeof(double),1,f)) s+=x;
}
long sonlar_miqdori=ftell(f)/sizeof (double);
s/=sonlar_miqdori; // s- o'rta arifmetik;
cout<<"Fayldagi sonlar o'rta arifmetigi="<<s<<'endl';
fseek(f,SEEK_SET,0); // fayl boshiga kelinsin;
int k=0;
while (fread(&x,sizeof(x),1,f))
{
    k+=(x<s); //o'rta arifmetikdan kichik elementlar soni;
}
fclose(f);
return k;
}
int main()
{
    if(Fayl_tuzish())
    {
        cout<<"Haqiqiy.son faylidagi \n";
        int Kichik=Kichiklar_soni();
        cout<<"O'rta arifmetikdan kichik sonlar miqdori=";
cout<<Kichik;
    }
    else //f faylini yaratish muvaffaqiyatsiz bo'ldi.
        cout<<"Haqiqiy.son faylini ochish imkoni bo'lmadi!!!";
return 0;
}
```

Dasturda bosh funksiyadan tashqari ikkita funksiya aniqlangan:

Int Fayl_tuzish() - diskda "Haqiqiy.son" nomli faylni yaratadi. Agar faylni yaratish muvaffaqiyatli bo'lsa, funksiya 1 qiymatini, aks holda 0 qiymatini qaytaradi. Faylni yaratishda klaviaturadan sonlarning satr ko'rinishi o'qiladi va haqiqiy songa aylantirilib, faylga yoziladi. Agar bo'sh satr kiritilsa, sonlarni kiritish jarayoni to'xtiladi va fayl yopiladi;

int Kichiklar_soni() funksiyasi diskdagi "Haqiqiy.son" nomli faylni o'qish uchun ochadi va fayl elementlarining o'rta arifmetigi s hisoblanadi so'ngra o'rta arifmetikdan kichik bo'lgan elementlar miqdori k hisoblanib, funksiya natijasi sifatida qaytariladi.

Bosh funksiyada faylning yaratilishi tekshiriladi va unga mos xabar beriladi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 7.001 Fayl nomi va $n(n > 1)$ butun soni berilgan. O'zida 1 dan n gacha bo'lgan sonlarni saqlovchi berilgan nomdagi fayl hosil qilinsin.
- 7.002 Fayl nomi va haqiqiy a , d sonlari berilgan. Tashqi faylga 1-hadi a ga, ayirmasi d ga teng bo'lgan arifmetik progressiyaning dastlabki n ta hadining qiymatlari yozilsin.
- 7.003 4 ta faylning nomi berilgan. Joriy katalogda joylashgan shu nomlar bilan berilgan fayllar soni topilsin.
- 7.004 Butun sonlar joylashgan fayl nomi berilgan. Fayldagi sonlar miqdori topilsin. Agar bunday nomdagi fayl topilmasa -1 chiqarilsin.
- 7.005 k butun soni va o'zida manfiy bo'lmagan butun sonlarni saqlovchi fayl berilgan. Fayldagi k -son chiqarilsin. Bunday son topilmasa -1 chop etilsin.
- 7.006 To'rttadan kam bo'lmagan songa ega butun sonli fayl berilgan. Ushbu fayldagi 1-, 2-, $n-1$, n -sonlar chop etilsin.
- 7.007 2 ta fayl nomi berilgan. Ulardan biri mavjud va o'zida haqiqiy sonlarni saqlaydi, 2-si esa joriy katalogda yo'q. 2-fayl tuzilsin va 1-fayldagi 1- va oxirgi sonlar nusxasi unga o'tkazilsin.
- 7.008 Haqiqiy sonli 2 ta fayl nomi berilgan. Ulardan biri mavjud va bo'sh emas, 2-si joriy katalogda yo'q. 2-fayl tuzilsin va 1-fayldagi oxirgi va 1- element nusxasi unga o'tkazilsin.
- 7.009 Butun sonlar joylashgan fayl berilgan. Shunday yangi fayl tuzish kerakki, tuzilgan faylda 1-fayldagi sonlar teskari tartibda joylashsin.
- 7.010 Haqiqiy sonlar joylashgan fayl berilgan. 2 ta yangi fayl tuzilsin: 1-faylga berilgan fayldagi toq o'rindagi sonlar, 2-faylga esa berilgan fayldagi juft o'rindagi sonlar joylashtirilsin.
- 7.011 Haqiqiy sonlar joylashgan fayl berilgan. 2 ta yangi fayl tuzilsin: 1-faylga berilgan fayldagi musbat sonlar, 2-faylga esa berilgan fayldagi manfiy sonlar joylashtirilsin.
- 7.012 Haqiqiy sonlar joylashgan fayl berilgan. Undagi elementlarning o'rta arifmetigi topilsin.
- 7.013 Haqiqiy sonlar joylashgan fayl berilgan. Fayldagi juft o'rinda joylashgan sonlari yig'indisi hisoblansin.
- 7.014 Butun sonlar joylashgan fayl berilgan. Undagi sonlar seriyalari miqdori hisoblansin.

- 7.015** Butun sonlar joylashgan fayl berilgan. Yangi fayl tuzilsin va unga berilgan fayldagi sonlar seriyalarining uzunliklari yozilsin.
- 7.016** Haqiqiy sonlar joylashgan fayl berilgan. Undagi sonlarning 1-lokal minimumi topilsin.
- 7.017** Haqiqiy sonlar joylashgan fayl berilgan. Undagi sonlarning oxirgi lokal maksimumi topilsin.
- 7.018** Haqiqiy sonlar joylashgan fayl berilgan. Fayldagi sonlarning barcha lokal ekstremumlari soni topilsin.
- 7.019** Haqiqiy sonlar joylashgan fayl berilgan. Yangi fayl tuzilsin va unga berilgan fayldagi lokal maksimumlarning tartib nomerlari joylashtirilsin.
- 7.020** Haqiqiy sonlar joylashgan fayl berilgan. Yangi fayl tuzilsin va unga berilgan fayldagi lokal extrimumlarning tartib nomerlari kamayish tartibida joylashtirilsin.
- 7.021** Haqiqiy sonlar joylashgan fayl berilgan. Yangi fayl tuzilsin va unga berilgan fayldagi kamayuvchi qisman ketma-ketliklarning uzunliklari chop etilsin.
- 7.022** Haqiqiy sonlar joylashgan fayl berilgan. Yangi fayl tuzilsin va unga berilgan fayldagi monoton qisman ketma-ketliklarning uzunliklari joylashtirilsin.
- 7.023** Haqiqiy sonlar joylashgan fayl berilgan. Undagi har bir son o'zining kvadrati bilan almashtirilsin.
- 7.024** Haqiqiy sonlar joylashgan fayl berilgan. Undagi eng katta va eng kichik elementlari o'zni almashtirilsin.
- 7.025** n ta butun sonlar joylashgan fayl berilgan. Fayldagi sonlarning joylashuv o'rinlari quyidagicha almashtirilsin:
 a_1 bilan a_n , a_2 bilan a_{n-1} ,
- 7.026** Haqiqiy sonlar joylashgan fayl berilgan. Undagi birinchi va oxirgi sonlaridan tashqari har bir sonning o'zidan oldingi va o'zidan keyingi son bilan tashkil etgan o'rta arifmetigi hisoblansin.
- 7.027** 5 tadan ko'p butun sonni saqlaydigan fayl berilgan. Fayldagi sonlarining dastlabki 5 ta sonidan tashqari sonlar o'chirilsin.
- 7.028** Juft miqdordagi butun sonlarni saqlaydigan fayl berilgan. Undagi sonlarning 2-yarmi o'chirilsin.
- 7.029** 5 tadan ko'p butun sonni saqlaydigan fayl berilgan. Fayldagi sonlarning dastlabki 5 tasi o'chirilsin.
- 7.030** Juft miqdordagi butun sonlarni saqlaydigan fayl berilgan. Undagi sonlarning 1-yarmi o'chirilsin.

- 7.031 Butun sonlar joylashgan fayl berilgan. Uning juft o‘rindagi barcha sonlar o‘chirilsin.
- 7.032 Butun sonlar joylashgan fayl berilgan. Undagi barcha manfiy sonlar o‘chirilsin.
- 7.033 50 tadan kam butun sonni saqlaydigan fayl berilgan. Undagi 1-son oldidan fayldagi sonlari miqdori 50 taga yetguncha 0 lar joylashtirilsin.
- 7.034 Butun sonlar joylashgan fayl berilgan. Uning sonlarini fayl oxiridan (berilgan tartibda) qayta yozish natijasida fayl o‘lchami 2 marta oshirilsin.
- 7.035 Butun sonlar joylashgan fayl berilgan. Uning sonlarini fayl oxiridan (teskari tartibda) qayta yozish natijasida fayl o‘lchami 2 marta oshirilsin.
- 7.036 Butun sonlar joylashgan fayl berilgan. Uning toq o‘rindagi barcha sonlar ikkilantirilsin.
- 7.037 Butun sonlar joylashgan fayl berilgan. Uning juft o‘rindagi har bir soni 2 ta 0 bilan almashtirilsin.
- 7.038 Butun sonlar joylashgan fayl berilgan. Uning har bir musbat soni 3 ta 0 bilan almashtirilsin.
- 7.039 Bir xil miqdordagi butun sonlarni saqlovchi 3 ta fayl berilgan. Har bir fayldan navbati bilan 1 tadan sonni o‘qib olib, yangi 4-fayl hosil qilinsin.
- 7.040 O‘shish tartibida joylashgan haqiqiy sonlarni saqlovchi 2 ta fayl berilgan. Ulardagi ma‘lumotlar yangi 3-faylga o‘shish tartibida joylashtirilsin.
- 7.041 O‘zida kamayish tartibida joylashgan haqiqiy sonlarni saqlovchi uchta fayl berilgan. Bu fayllardagi sonlar to‘rtinchi fayliga kamayish tartibida joylashtirilsin.
- 7.042 n ta ($n \leq 4$) fayl berilgan. Yangi arxiv faylga berilgan fayllardagi ma‘lumotlar quyidagicha biriktirilsin: Dastlab berilgan faylar soni n , so‘ngra mos faylar o‘lchamini bildiruvchi n ta son, keyin har bir berilgan fayldagi ma‘lumotlar tartib bilan yoziladi.
- 7.043 n ta ($n \leq 4$) fayl berilgan. Fayllardagi saqlanayotgan ma‘lumotlar yangi fayl arxivga quyidagi ko‘rinishda biriktirilsin: 1-berilgan faylning o‘lchami va undagi ma‘lumotlar, so‘ngra 2-berilgan faylning o‘lchami va undagi ma‘lumotlar va hokazo.
- 7.044 Hech bo‘lmaganda 1 ta bo‘sh joyni saqlaydigan fayl berilgan. Bu fayldan bo‘sh joydan keyingi, 1-belgidan tashqari barcha belgilari o‘chirilsin.

- 7.045** Hech bo‘limganda 1 ta bo‘sh joyni saqlaydigan fayl berilgan. Bu fayldagi bo‘sh joy va undan oldingi 1-belgidan tashqari barcha belgilari o‘chirilsin.
- 7.046** Hech bo‘limganda 1 ta bo‘sh joyni saqlaydigan fayl berilgan. Oxirgi bo‘sh joydan oldindagi belgidan tashqari barcha belgilar o‘chirilsin.
- 7.047** $k(k>0)$ butun soni va har biri satrida k tadan kam bo‘lmagan belgini saqlovchi fayl berilgan. 2 ta yangi fayl tuzilsin: 1-siga berilgan faylning har bir satridagi k tadan belgi, 2-siga berilgan faylning har bir satridagi k -belgilari saqlansin.
- 7.048** Satrlarni saqlovchi fayl berilgan. Berilgan fayldagi satrlar miqdoridan kichik uzunlikka ega bo‘lgan barcha satrlar chop etilsin.
- 7.049** Satrlarni saqlovchi fayl berilgan. Berilgan fayldagi satrlar miqdoridan katta uzunlikka ega bo‘lgan barcha satrlar chop etilsin.
- 7.050** Satrlarni saqlovchi fayl berilgan. Berilgan fayldagi barcha satrlar leksikografik (lug‘at) tartibda joylashtirib, yangi faylga yozilsin.
- 7.051** “Kun/oy/yil” sana formatidagi ma‘lumotlarni saqlovchi fayl berilgan. 2 ta yangi fayl hosil qilinsin, 1-siga berilgan fayldagi sana ma‘lumotining kun qiymati, 2-ga esa oy qiymati yozilsin.
- 7.052** “Kun/oy/yil” sana formatidagi ma‘lumotlarni saqlovchi fayl berilgan. Berilgan fayldagi barcha yozgi sanalarni o‘zida saqlovchi yangi fayl hosil qilinsin.
- 7.053** “Kun/oy/yil” sana formatidagi ma‘lumotlarni saqlovchi fayl berilgan. Berilgan fayldagi sanalar kamayish tartibida tartiblanib, yangi faylga yozilsin.
- 7.054** i, j butun sonlarni va haqiqiy sonlardan tashkil topgan kvadrat matritsani saqlovchi fayl berilgan. Fayldagi matritsaning i -satri va j -ustunida joylashgan son chop etilsin. Agar bunday son yo‘q bo‘lsa, 0 chiqarilsin.
- 7.055** Haqiqiy sonlardan tashkil topgan kvadrat matritsani saqlovchi fayl berilgan. Berilgan fayldagi matritsaning transponerlangan holatini saqlovchi yangi fayl tuzilsin.
- 7.056** Haqiqiy sonlardan tuzilgan a va b kvadrat matritsalar mos ravishda 2 ta faylda joylashgan. Berilgan fayllardagi matritsalarining ko‘paytmasini o‘zida saqlovchi yangi fayl tuzilsin. Agar fayllardagi matritsalarini ko‘paytirish mumkin bo‘lmasa fayl bo‘sh qoldirilsin.

- 7.057** i, j butun sonlari va haqiqiy sonlardan tashkil topgan to'g'ri to'rtburchakli matritsani saqlaydigan fayl berilgan. Faylning birinchi satrida matritsa ustunlari soni joylashadi. Fayldagi matritsaning i -satr va j -ustunida joylashgan son chiqarilsin. Agar bunday son yo'q bo'lsa, 0 chiqarilsin.
- 7.058** Haqiqiy sonlardan tashkil topgan to'g'ri to'rtburchakli matritsani saqlaydigan fayl berilgan. Faylning birinchi satrida matritsa ustunlari soni joylashadi. Berilgan fayldagi matritsaning transponerlangan holatini saqlovchi yangi fayl tuzilsin.

7.2. Matnli fayllar

Matnli fayl—ASCII kodidagi belgilar bilan berilganlar majmuasidir. Belgilar ketma-ketligi satrlarni tashkil etadi va satrning tugash alomati CR (karetkani qaytarish yoki '\r'), LF (satrni o'tkazish yoki '\n') belgilar juftligi hisoblanadi. Matn fayldan berilganlarni o'qishda bu belgilar juftligi bitta belgiga - CR belgisiga almashtiriladi va aksincha, yozishda CR belgisi ikkita CR va LF belgilariga almashtiriladi. Fayl oxiri #26 (^Z) belgisi bilan belgilanadi.

Matn faylga boshqacha ta'rif berish ham mumkin. Agar fayldagi ma'lumotni matn tahrirlagichlari yordamida ekranga chiqarib o'qish mumkin bo'lsa, bunday fayllar *matn fayllari* deyiladi (Klaviatura ham kompyuterga faqat matnlarni jo'natadi). Boshqacha aytganda dastur tomonidan ekranga chiqariladigan barcha ma'lumotlarni *stdout* nomidagi matn fayliga chiqarilayapdi deb qarash mumkin. Xuddi shunday klaviaturadan o'qilayotgan ixtiyoriy berilganlar matn faylidan o'qilayapdi deb hisoblanadi.

Matn fayllarining komponentalari satrlar hisoblanadi. Satrlar uzluksiz joylashishi, turli uzunlikda va bo'sh bo'lishi mumkin.

Matnli fayllar bilan ishlash uchun ham *binar* fayllari bilan ishlashdagi funksiyalardan foydalanish mumkin, ammo matnli fayllarning o'ziga xos tushunchalar ham mavjud. Masalan, matnli fayl ochilayotganligini bildirish uchun fayl ochilish rejimi satriga 't' belgisini qo'shib yozish zarur bo'ladi. Misol uchun fayl ochilishining "rt+" rejimi matnli faylning o'qish uchun ochilganligini bildiradi.

Faraz qilaylik "D:\KUTUBXONA\Prog_mas.txt" nomli matn faylini o'qish uchun ochish zarur bo'lsin. Bu talab,

```
FILE * kitobf=fopen("D:\KUTUBXONA\Prog_mas.txt", "rt+");
```

ifodasini yozish orqali amalga oshiriladi. Natijada diskda mavjud bo'lgan fayl dasturda «*kitobf*» o'zgaruvchisi nomi bilan aynan bir xil deb tushuniladi. Boshqacha qilib aytganda, dasturda keyinchalik «*kitobf*» o'zgaruvchisi ustida bajarilgan barcha amallar, «*D*» diskdagi «*Prog_mas.txt*» fayli ustida ro'y beradi.

Quyida keltirilgan dasturda ma'lumotlarni faylga yozish va undan o'qishga doir misollar keltirilgan.

```
#include <iostream.h>
#include <stdio.h>
int main()
{
    char c;
    FILE *f1,*f2; // fayl turidagi o'zgaruvchilarni e'lon qilish;
    // f1 va f2 o'zgaruvchilarga bog'langan fayllarni ochish;
    if((f1=fopen("D:\Kitoblar\Prog_mas.doc","rt+"))==NULL)
    {
        cout<<"Prog_mas.doc fayli ochilmadi!!\n";
        return 1;
    }
    if((f2=fopen("E:\Hujjatlar\xulosa.txt","wt+"))==NULL)
    {
        cout<<"xulosa.txt fayli ochilmadi!!\n";
        return 1;
    }
    while (!feof(f1))
    {
        char c=fgetc(f1); /* "f1" o'zgaruvchisiga bog'langan fayldan
        ma'lumotlarni "c" o'zgaruvchisiga o'qish; */
        cout<<c; // "c" o'zgaruvchisidagi belgini ekranga chiqarish;
        fputc(c,f2); /* "c" o'zgaruvchisidagi ma'lumotni "f2"
        o'zgaruvchisi bilan bog'langan faylga yozish; */
    }
    // "f1" va "f2" o'zgaruvchilari bilan bog'langan fayllarni yopish;
    fclose(f1);
    fclose(f2);
    return 0;
}
```

Dasturda «*Prog_mas.doc*» fayli matn fayli sifatida o'qish uchun ochilgan va u «*f1*» o'zgaruvchisi bilan bog'langan. Xuddi shunday,

“xulosa.txt” matn fayli yozish uchun ochilgan va “f2” o’zgaruvchisi bilan bog’langan. Agar faylni ochish muvaffaqiyatsiz bo’lsa, mos xabar beriladi va dastur o’z ishini tugatadi. Keyinchalik, “f1” ga xizmat qilayotgan fayl ko’rsatgichi fayl oxiriga yetmaguncha, “f1” ga bog’langan fayldan belgilarni o’qish davom etadi va o’qilgan belgilar bir vaqtning o’zida ekranga hamda “f2” bilan bog’langan tashqi faylga chiqariladi. Dastur oxirida ikkita fayl ham yopiladi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 7.059 Fayl nomi va butun musbat n va k sonlari berilgan. Ko’rsatilgan nomdagi matn fayli tuzilsin va uning n ta satrining har biriga k tadan «*» yozilsin.
- 7.060 Fayl nomi va butun n ($0 < n < 27$) soni berilgan. Ko’rsatilgan nomdagi matn fayl tuzilsin va unga n ta satr quyidagi ko’rinishda yozilsin: 1-satrga «a», 2-satrga «ab», 3-satrga «abc» va hokazo, n satrga n ta kichik lotin alifbosi harflari alfavit tartibida yozilsin.
- 7.061 Fayl nomi va butun n ($0 < n < 27$) soni berilgan. Ko’rsatilgan nomdagi matn fayl tuzilsin va unga n ta satr quyidagi ko’rinishda yozilsin: Har bir k -satrda n tadan belgi bo’lib, belgilarning k tasi alfavit tartibidagi lotin bosh harflari qolganlari «*» belgisidan iborat. Misol: «a***», «ab**», «abc*», «abcd». $n=4$.
- 7.062 Matnli fayl berilgan. Unda joylashgan belgilar va satrlar soni chiqarilsin.
- 7.063 Satr va matnli fayl berilgan. Fayl oxiriga berilgan satr qo’shilsin.
- 7.064 2 ta matnli fayl berilgan. 1-fayl oxiriga 2-faylda saqlanayotgan ma’lumotlar qo’shilsin.
- 7.065 Satr va matnli fayl berilgan. Fayl boshiga berilgan satr qo’shilsin.
- 7.066 2 ta matnli fayl berilgan. 1-fayl boshiga 2-faylda saqlanayotgan ma’lumotlar qo’shilsin.
- 7.067 k butun soni va matnli fayl berilgan. Faylning k -satridan oldin yangi bo’sh satr qo’shilsin. Agar faylda k -satr bo’lmasa fayl o’zgartirilmasin.
- 7.068 k butun soni va matnli fayl berilgan. Faylning k -satridan keyin yangi bo’sh satr qo’shilsin. Agar faylda k -satr bo’lmasa fayl o’zgartirilmasin.
- 7.069 Matnli fayl berilgan. Uning barcha bo’sh satrlari ikkilantirilsin.

- 7.070** Satr va matnli fayl berilgan. Faylning barcha bo'sh satrlari berilgan satr bilan almashtirilsin.
- 7.071** Bo'sh bo'lmagan matnli fayl berilgan. Uning 1-satri o'chirilsin.
- 7.072** Bo'sh bo'lmagan matnli fayl berilgan. Uning oxirgi satri o'chirilsin.
- 7.073** k butun soni va matnli fayl berilgan. Faylning k -satri o'chirilsin.
- 7.074** Matnli fayl berilgan. Uning barcha bo'sh satrlari o'chirilsin.
- 7.075** 2 ta matnli fayl berilgan. 1-faylning har bir satrining oxiriga 2-faylning mos satrlari qo'shilsin.
- 7.076** k butun soni va matnli fayl berilgan. Faylning har bir satridagi boshlang'ich k ta belgi o'chirilsin.
- 7.077** Matnli fayl berilgan. Fayldagi barcha lotin bosh harflar kichik harflar bilan va aksincha almashtirilsin.
- 7.078** Matnli fayl berilgan. Fayldagi yonma-yon kelgan bir nechta bo'sh joy bitta bo'sh joy bilan almashtirilsin.
- 7.079** 3 ta dan kam bo'lmagan satrga ega matnli fayl berilgan. Uning oxirgi 3 ta satri o'chirilsin.
- 7.080** k ($0 < k < 10$) butun son va k tadan kam bo'lmagan satrga ega matnli fayl berilgan. Uning oxirgi k ta satri o'chirilsin.
- 7.081** k ($0 < k < 10$) butun son va k tadan kam bo'lmagan satrga ega matnli fayl berilgan. Berilgan faylning oxirgi k ta satrini saqllovchi yangi matnli fayl tuzilsin.
- 7.082** Matnli fayl berilgan. Matndagi abzatslar soni topilsin.
- 7.083** k butun soni va matnli fayl berilgan. Fayldan k -nomerdagi abzats o'chirilsin. O'chirilgan abzats joyi bo'sh qolmasin. k -nomerdagi abzats topilmasa fayl o'zgarishsiz qoldirilsin.
- 7.084** Matnli fayl berilgan. Matndagi abzatslar soni hisoblansin.
- 7.085** Matnli fayl berilgan. Matndagi har bir abzatsning boshlang'ich satrida 5 tadan bo'sh joy bo'lgan abzatslar soni hisoblansin.
- 7.086** k butun son va matnli fayl berilgan. Fayldan k -nomerdagi abzats o'chirilsin. O'chirilgan abzats joyi bo'sh qolsin. k -nomerdagi abzats topilmasa fayl o'zgarishsiz qoldirilsin.
- 7.087** Matnli fayl berilgan. Matndagi abzatslar chiziqli satr bilan ajratilgan. Abzatslar orasiga 1 tadan bo'sh satr joylashtirilsin.
- 7.088** Matnli fayl berilgan. Matnda 1-uchragan eng uzun so'z chiqarilsin.
- 7.089** Matnli fayl berilgan. Matndagi oxirgi uchragan eng qisqa so'z uzunligi chiqarilsin.
- 7.090** k butun soni va har bir satrida k tadan kam bo'lmagan belgini

saqlovchi matnli fayl berilgan. Berilgan faylning har bir satridan k tadan belgi ajratib olinib, yangi faylga yozilsin.

B toifa

- 7.091** Lotin bosh harflaridan birini ifodalaydigan belgi va matnli fayl berilgan. Berilgan matnli fayldagi shu belgi bilan boshlanuvchi satrlar nusxasi yangi faylga o'tkazilsin.
- 7.092** Lotin kichik harflaridan birini ifodalaydigan belgi va matnli fayl berilgan. Berilgan fayldagi hech bo'lmaganda 1 ta shu belgi uchragan satr yangi faylga o'tkazilsin.
- 7.093** Chap tomoni bo'yicha tartiblangan matnni saqlovchi fayl berilgan. Chap tomoniga keraklicha bo'sh joy qo'shish yordamida matn o'ng tomon bo'yicha tartiblansin (Satr uzunligi 50 ga teng).
- 7.094** Chap tomoni bo'yicha tartiblangan matnni saqlovchi fayl berilgan. Chap tomoniga keraklicha bo'sh joy qo'shish yordamida matn satrlari markazlashtirilsin (Satr uzunligi 50 ga teng).
- 7.095** O'ng tomoni bo'yicha tartiblangan matnni saqlovchi fayl berilgan. Boshlang'ich bo'sh joylarning yarmini olib tashlash evaziga matn markazlashtirilsin.
- 7.096** Chap tomoni bo'yicha tartiblangan matnni saqlovchi fayl berilgan. Matn abzatsi boshlangan satr 1 ta bo'sh joy bilan farq qiladi. So'zlar o'rtasidagi bo'sh joylarni oshirish evaziga matnning har bir satri kengligi bo'yicha tartiblansin.
- 7.097** k ($k > 25$) butun son va chap tomon bo'yicha tartiblangan matnni saqlovchi fayl berilgan. Matndagi har bir abzats bitta bo'sh joy bilan boshlangan. Matn quyidagi tartibda formatlanib, yangi faylga yozilsin: satr uzunligi k dan oshmasa abzats belgisini saqlagan holda satr kengligi bo'yicha tartiblansin, satr oxiridagi bo'sh joy o'chirilsin.
- 7.098** a, b haqiqiy sonlar va n butun soni berilgan. $\sqrt{x}$ funksiyaning $[a, b]$ oraliqdagi $\frac{|b-a|}{n}$ qadam bilan hisoblangan qiymatlari jadvalini saqlovchi fayl tuzilsin.
- 7.099** a, b haqiqiy sonlar va n butun soni berilgan. $\sin(x)$ va $\cos(x)$ funksiyalarning $[a, b]$ oraliqdagi $\frac{|b-a|}{n}$ qadam bilan hisoblangan qiymatlari jadvalini saqlovchi fayl tuzilsin.

- 7.100 Har bir satrida butun sonlarni saqlovchi fayl berilgan. Bu fayldagi sonlarning miqdori va yig'indisi aniqlansin.
- 7.101 Har bir satrida haqiqiy sonlarni saqlovchi fayl berilgan. Kasr qismi 0 bo'lmagan sonlar miqdori va ularning yig'indisi chiqarilsin.
- 7.102 Har bir satrida haqiqiy sonlarni saqlovchi fayl berilgan. Berilgan fayldan kasr qismi 0 bo'lmagan barcha haqiqiy sonlarni ajratib olib, yangi faylga yozilsin.
- 7.103 Har bir satrida bo'sh joylar bilan ajratilgan 1 nechta sonlar tasvirlangan fayl berilgan. Berilgan fayldagi barcha butun sonlarni saqlovchi yangi fayl tuzilsin.
- 7.104 Birinchisi matnni va ikkinchisi esa butun sonlarni saqlovchi 2 ta fayl berilgan. Birinchi faylning har bir satri oxiriga ikkinchi fayldan 1 tadan son olib qo'shilsin.
- 7.105 Fayl berilgan. Fayldagi har bir satr boshida 10 ta harf qolgan qismi esa sonlardan iborat. 2 ta yangi fayl tuzilsin: 1-faylning satrlarida berilgan faylning mos satrlaridagi harflar, 2-faylning satrlarida esa berilgan faylning mos satrlaridagi sonlar joylashtirilsin.
- 7.106 3 ta ustundan iborat haqiqiy sonlar jadvalini saqlovchi fayl berilgan. Ustunlarni ajratuvchi belgini hisobga olib shunday 3 ta fayl hosil qilinsin va ularning har biriga berilgan fayldagi 1 tadan ustun yozilsin.
- 7.107 3 ta ustundan iborat butun sonlar jadvalini saqlovchi fayl berilgan. Ustunlarni ajratuvchi belgini hisobga olib shunday, 3 ta fayl hosil qilinsin va ularning har biriga berilgan fayldagi 1 tadan ustun yozilsin.
- 7.108 Fayl berilgan. Berilgan fayldagi matnda uchraydigan barcha tinish belgilarini saqlovchi yangi fayl tuzilsin.
- 7.109 Fayl berilgan. Berilgan fayldagi bo'sh joy va tinish belgilaridan tashqari barcha belgilarni saqlovchi yangi fayl tuzilsin.
- 7.110 Fayl berilgan. Berilgan fayldagi bo'sh joy va tinish belgilaridan tashqari barcha belgilarning kodini saqlovchi fayl tuzilsin.
- 7.111 Fayl berilgan. Har bir kichik lotin harfini satrda joylashgan o'rni bilan « $\langle harf \rangle - \langle son \rangle$ » ko'rinishida tasvirlansin. Misol uchun: «a-25».
- 7.112 10 ta raqamdan iborat s satr va matnli fayl berilgan. Yuqorida berilgan satrdagi raqamlar yordamida matn shifrlansin: k -o'rindagi belgi o'zining alfavitdagi o'rnidan s_k pozitsiya keyingi

belgi bilan almashtirilsin, bo'sh joy va tinish belgilar o'zgarishsiz qoldirilsin. Agar harf alfavitning oxirgi s_k ta elementlaridan biri bo'lsa, u "*harf kodi*" + $s_k - 26$ o'rindagi harf bilan almashtirilsin. Agar $k=11$ bo'lsa shifrlash yana s_1 dan boshlansin va hokazo.

- 7.113** Yuqorida berilgan masaladagi usul bilan shifrlangan matnli fayl va satr berilgan. Berilgan satr shifrlangan matnning asl holidagi birinchi satrini ifodalaydi. Shundan foydalanib, matn o'z holiga qaytarilsin. Agar ma'lumot yetarli bo'lmasa shifrlangan fayl o'zgarishsiz qoldirilsin.

7.3. Bobga doir qiyinroq masalalar

Berilganlar tashqi fayldan o'qiladi, natija ham tashqi faylga yoziladi.

- 7.114** *Tort [12].*

Dinazavr bolasi bugun n yoshga to'ladi. U uyiga bozordan radiusi r bo'lgan aylana shaklidagi tort olib keldi. Tortda n ta sham bor. Dinazavrning onasi tortga m marta to'g'ri chiziq qilib pichoq tortdi. Dinazavrning uyiga kelgan mehmonlarning har biri bir bo'lakdan tort yeyishdi. Dinazavr mehmonlardan kimgadir ikkita shamli bo'lak uzatildimi, yo'qmi? bilishni hohlayapti. Unga yordam bering.

Berilganlar:

Birinchi qatorda n, m va r butun sonlari ($1 \leq n \leq 1000$, $0 \leq m \leq 500$, $1 \leq r \leq 500$). Aylana markazi koordinatalar boshida yotadi. Keyingi n ta qatorda har bir sham koordinatalari – x_i, y_i sonlar bilan beriladi. Ular aylana ichida yotishi va ustma-ust tushmasligi kafolatlanadi. Oxirgi m ta qatorda tortni kesuvchi to'g'ri chiziq tenglamasi koeffitsientlari – a_i, b_i, c_i butun sonlari. Bu uchlik $a_i x + b_i y + c_i = 0$ to'g'ri chiziq tenglamasini qanoatlantiradi (a, b, c sonlar bir vaqtda nolga teng emas). To'g'ri chiziqlar shamlarning ustidan kesib o'tmasligi va ixtiyoriy ikki chiziq ustma-ust tushmasligi kafolatlanadi. a_i, b_i, c_i sonlar modul jihatdan 100 dan oshmaydi.

Natija:

Agar tortning qaysidir bo'lagida bittadan ko'p sham qolgan bo'lsa "YES" so'zini chiqaring, aks holda "NO" so'zini chiqaring.

<u>Berilganlar:</u>	<u>Natija:</u>
$3\ 2\ 3$ $2\ 2$ $1\ -1$ $-2\ 0$ $2\ -1\ 0$ $0\ 1\ -1$	NO
$3\ 2\ 3$ $2\ 2$ $1\ -1$ $-2\ 0$ $1\ 1\ -1$ $0\ 1\ -1$	YES
$1\ 0\ 100$ $0\ 0$	NO

7.115

Qiziqarli masala [12].

Dasturchilar maktabi o'qituvchisi qiziqarli masalalarni yoqtiradi U o'quvchilarga qiziqarli topshiriq berdi, ya'ni ularga o'z tug'ilgan sanasiga 5 ni qo'shib, 50 ga ko'paytirishni aytdi. So'ngra tug'ilgan oyni qo'shib joriy yilni (*ya'ni 2015-yilni*) ko'paytirishni va tug'ilgan yilini qo'shishni aytdi. Chiqqan natijalarga qarab bolalarning qaysi yilda, oyda va qaysi kunda tug'ilganligini aytib berdi. Siz ham urinib ko'ring. O'quvchilar 1970-yil 1-yanvardan 2000 - yil 31 - dekabrigacha tug'ilgan deb hisoblansin. Kabisa yili inobatga olinsin. Kabisa yili deb 400 ga bo'linadigan yoki 4 ga bo'linib, 100 ga bo'linmaydigan yillar kiradi.

Berilganlar:

Birinchi qatorda n bolalar soni. Keyin n ta qatorda bolalarning natijalari X_i ($1 \leq n \leq 26$). ($1 \leq X_i \leq 2^{31} - 1$).

Natija:

N ta qatorda bolalarning tug'ilgan sanasi, oyi, yili nuqta bilan ajratilgan holatda chiqarilsin.

<u>Berilganlar:</u>	<u>Natija:</u>
3 3035098 2624226 5053092	xato hisoblangan 21.2.1998 xato hisoblangan
4 5349185 1931426 1013024 3429726	xato hisoblangan xato hisoblangan 5.2.1996 xato hisoblangan

7.116

Radioviktorina [12].

Shahzod radioviktorinalarni juda yaxshi ko'radi. U yaqinda yangi radioviktorinada ishtirok eta boshladi. Yangi radioviktorina shartiga asosan ishtirokchiga N ta son aytiladi, ishtirokchi bu sonlar orasidan $N/2$ martadan ko'p, ya'ni aytilgan sonlarning yarmidan ko'p marta takrorlangan sonni aytishi kerak. Mana necha martadirki, Shahzod bu viktorinada yuta olmayapti. Siz ushbu viktorinada yutiga yordam beruvchi dasturni tuzishingiz kerak.

Berilganlar:

Birinchi qatorda N natural son ($1 \leq N \leq 3 \cdot 10^6$) aytiladigan sonlar miqdori. Keyingi N ta qatorda a_i soni ($-10^{18} \leq a_i \leq 10^{18}$)

Natija:

Bitta butun son, berilgan sonlar orasida $N/2$ martadan ko'p takrorlangan son. (Sonlar yarmidan ko'pi bir xil bo'lishi kafolatlanadi).

<u>№</u>	<u>Berilganlar:</u>	<u>Natija:</u>
1	5 1 3 18 1	1

	1	
2	3 -18 0 -18	-18

7.117

Azkabandan qochish [12].

Afsungar *Sirius* afsungar *Garri Potter* yordamida Azkaban qamoqxonasidan qochmoqchi. *Garri Siriusga* cheklangan miqdorda yordam bera oladi xolos. Qamoqxona N ta bo'limdan iborat bo'lib, N chi bo'limda chiqish yo'li bor. *Sirius* hozir 0 chi bo'limda. *Garrining* afsuni yordamida *Sirius* turgan bo'limdan g'oyib bo'lib, K uzoqlikdagi bo'limlardan birida paydo bo'la oladi. *Garri* bir chetda bitta afsun yordamida *Siriusni*, u turgan bo'limdan K uzoqliqgacha bo'lgan bo'limlardan birida bo'lishini ta'minlaydi (Faqat o'zi turgan bo'limdan chiqish yo'li tomon K uzoqliqgacha, *Sirius* o'zi turgan bo'limida yana qaytadan bo'lishi yoki orqaga tomon o'tishi mumkin emas). Har bir bo'limda qo'riqchilar bo'lib, ular *Siriusning* sog'ligiga a_i ta'sir o'tkazadi. *Sirius* ayrim qo'riqchilar bilan yaxshi munosabatda bo'lgani uchun ular unga musbat ta'sir ko'rsatadi, ya'ni ular uni davolashadi. Qolgan qo'riqchilar esa unga manfiy ta'sir o'tkazadi. Siz *Garriga* shunday yordam beringki, *Sirius* qamoqxonaning N chi bo'limiga borganda uning sog'ligi eng yaxshi bo'lsin. Sizdan *Siriusning* N chi bo'limga borgandagi sog'ligi so'raladi.

Berilganlar:

Birinchi qatorda N va K natural sonlari ($1 \leq N \leq 10^6$, $1 \leq K \leq N$). Keyingi N ta qatorda a_i soni ($-10^9 \leq a_i \leq 10^9$), i -bo'limdagi qo'riqchining *Siriusga* ta'siri.

Izoh: Berilgan sonlar $[1; N]$ bo'limdagi qo'riqchilar ta'siri, *Sirius* esa 0 chi bo'limda turibdi.

Natija:

Bitta butun son, masala javobi.

<i>N_o</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	5 2 1 2 1 1 3	8
2	10 3 -4 -5 -6 -7 -8 -9 8 7 6 5	15
3	7 3 -1 -4 -3 -7 -6 -7 5	-3

1 – test uchun javob: Sirius 1,2,3,4,5 bo'limlardan o'tdi.

2 – test uchun javob: Sirius 1,4,7,8,9,10 bo'limlardan o'tdi.

3 – test uchun javob: Sirius 1,4,7 bo'limlardan o'tgan.

7.118

Fan imtihoni [12].

Fan imtihonlari vaqtida barcha talabalar harakatga tushib, qo'llariga kitob ola boshlaydi. Odatda imtihon bilet savollari oldindan beriladi, lekin bu biletning hammasiga tayyorlanish qiyin albatta. O'qituvchilar ham doimo bir xil, imtihon paytida biletlarni almashtirib tashlashadi, chunki yonma-yon o'tirgan talabalarga bir xil bilet tushib qolishi mumkin. Bugun ham talabalar imtihon topshirishlari kerak. Auditoriyada N ta talaba bitta qator bo'lib o'tirishibdi. O'qituvchi kelib ularga biletlarini tarqatganda i -talabaga a_i -bilet to'g'ri keldi (bir-nechta talabalarga bitta bilet tushishi ham mumkin). Endi bir-xil biletli talabalar birgalikda imtihon biletiga javob yozishni rejalashtirayapti, lekin o'qituvchi bunga yo'l qo'ymasligi tabiiy hol. Auditoriyada faqat o'qituvchi nazaridan qochib yonma-yon o'tirgan talabalar joy almasha olishadi (ya'ni, i -talaba $(i+1)$ -talaba bilan joy almasha oladi). Siz shunday dastur tuzingki, eng kam almashtirish orqali bir xil biletli talabalar bir joyda o'tirishsin.

Berilganlar:

Birinchi qatorda N natural soni ($1 \leq N \leq 10^4$). Keyingi qatorda N ta butun son, i -talabaga tushgan bilet raqami ($1 \leq a_i \leq 7$).

Natija:

Minimum almashtirishlar sonini chiqaring.

Berilganlar:	Natija:
6 1 2 1 3 2 2	2

1-test uchun izoh: bu yerda almashinishlardan keyin talabalar $\{1, 1, 3, 2, 2, 2\}$ ko'rinishda bo'ladi.

7.119 Ko'paytma oxiridagi nollar [12].

n ta elementdan iborat bo'lgan butun sonlar ketma-ketligi berilgan. Sizning vazifangiz bu ketma-ketlikdagi sonlarni bir-birida ko'paytirib, uning oxiri nechta nol bilan tugaganligini topish.

Berilganlar:

Birinchi qatorda ketma-ketlikdagi sonlar miqdori n ($1 \leq n \leq 10^5$) berilgan. Ikkinchi qatorda n ta butun son, ketma-ketlik elementlari berilgan. Ular absolyut qiymat jihatidan 10^9 dan oshmaydi.

Natija:

Birinchi qatorda ketma-ketlik elementlari ko'paytmasining nechta nol bilan tugashini chiqaring.

№	Berilganlar:	atija:
1	3 4 5 2	1
2	4 -5 4 50 8	3

7.120 Omadli contest [12].

Bugungi kunda dunyodagi dasturlash bo'yicha eng yaxshi sayt *Codeforces* sayti hisoblanadi (to'liq adresi: codeforces.ru). Unda doimiy musoboqalar o'tkaziladi va qatnashuvchilarga uning asosida reyting ballari beriladi. Komron Yusupov *codeforcesning* ashaddiy muxlisi. U deyarli barcha musoboqalarda qatnashadi. Uning uchun reytingi avvalgisiga nisbatan ko'tarilishi omadlicontest, tushishi yoki o'zgarmasligi esa omadsizcontest hisoblanadi. Sizga uning ma'lum oraliqda qatnashgan musoboqalardan keyingi reytinglari va bu

musoboqalarga qatnashishdan oldingi reytingi berilgan. Sizning vazifangiz Komron Yusupov uchun maksimal nechta ketma-ket uzluksiz omadlicontest bo'lganligini aniqlash.

Berilganlar:

Birinchi qatorda n qaralayotgan musoboqalar soni ($1 \leq n \leq 10^5$) va X musoboqalarga qatnashishdan oldingi reytingi beriladi. Ikkinchi qatorda n ta butun son–har bir musobaqadan keyingi reytingi, X soni va har bir reyting butun sondan iborat bo'lib, ular 0 dan 3200 gacha oraliqda bo'lishi mumkin.

Natija:

Omadlicontestlar soni chiqarilsin.

<i>№</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	12 1678 1736 1770 1789 1747 1762 1739 1728 1746 18281868 1914 1823	4
2	4 2010 1931 1859 1823 1765	0
3	1 1716 1723	1

Izoh: 1-misolda uzluksiz omadlicontestlar : (1, 2, 3), (5), (8, 9, 10, 11). Demak maksimal ketma-ket 4 ta omadlicontest bo'lgan.

7.121

Fillar [12].

$n \times n$ shaxmat doskasi berilgan bo'lib, unda faqat fillar joylashgan. Fil standart shaxmat qoidasiga ko'ra u bilan bir diogonalda turgan kataklarga hujum qila oladi. Rasmda 'F'-fil turgan katak. '1' lar esa uning ta'qibi ostida turgan kataklar.

				1		1	
					F		
				1		1	
			1				1
		1					
	1						
1							

Berilganlar:

Birinchi qatorda n butun soni –taxtaning o'lchami berilgan ($1 \leq n \leq 40$). Keyingi n ta qatorning har birida n ta belgi bittadan bo'sh joy bilan ajratib berilgan. U belgilar '0' yoki 'F' bo'lishi mumkin. '0' bo'lsa bo'sh, 'F' bo'lsa fil turgan deb hisoblanadi.

Natija:

n ta qatorning har birida n tadan natijaviy belgilar bittadan bo'sh joy bilan ajratib chiqariladi. Fillar turgan kataklar 'F', ta'qib ostidagi kataklar esa '1', qolgan kataklar '0' ligicha chiqarilsin.

<i>Nº</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	4 0 F 0 0 F 0 F 0 0 0 0 0 0 0 0 0	0 F 0 1 F 0 F 0 0 1 0 1 1 0 1 0
2	1 0	0

VIII Bob. Funktsiyalarda ma'lumotlarning murakkab turlari

8.1. Bir va ikki o'lchovli massivlar bilan ishlashda funktsiyalardan foydalanish

C++ tilida funktsiyalar massivlarni parametr sifatida qabul qilishi va ularni natija sifatida qaytarishi mumkin.

Agar massiv parametr orqali funktsiyaga uzatilsa, massiv elementlari sonini aniqlash masalasi tug'iladi, chunki massiv nomidan foydalanib, undagi elementlar sonini aniqlab bo'lmaydi. Belgilar massivi sifatida aniqlangan satr (*ASCIIz satrlar*) bilan ishlaganda esa massiv uzunligini aniqlash mumkin bo'ladi, chunki satrlar '\0' belgisi bilan tugallanadi.

Funktsiya parametrlari belgilar massivi (satr) bo'lmagan hollarda fiksirlangan uzunlikdagi massivlar ishlatiladi. Agar turli uzunlikdagi massivlarni uzatish zarur bo'lsa, massiv o'lchamlarini parametr sifatida uzatish mumkin yoki bu maqsadda global o'zgaruvchidan foydalanishga to'g'ri keladi.

Masala. Elementlari o'sish tartibida tartiblangan 2 ta butun sonli massivlar berilgan. Bu massivlardagi elementlarning barchasi yangi massivga o'tkazilsin va tashkil etilgan massiv elementlarining o'sish tartibida bo'lishi ta'minlansin.

```
#include <iostream.h>
// butun turdagi massivga ko'rsatkich qaytaradigan funktsiya
int *massivlarni_birlashtirish(int,int*,int,int*);
void main()
{
    int c[]={-4,7,18,21,85},d[]={3,8,74};
    int *h;
    h=massivlarni_birlashtirish(5,c,3,d);
    for(int i=0;i<8;i++) cout<<"\t"<<h[i];
    delete[]h;
}
int *massivlarni_birlashtirish(int n,int *a,int m,int *b);
{
    int *x=new int[n+m];
    int ia=0,ib=0,ix=0;
    while (ia<n && ib<m)
```



```

    a[ia]>b[ib]?x[ix++]=b[ib++]:x[ix++]=a[ia++];
    while (ib<m) x[ix++]=b[ib++];
    while (ia<n) x[ix++]=a[ia++];
    return x;
}

```

Ko'p o'lchamli massivlar bilan ishlash ma'lum bir murakkablikka ega, chunki bunday massivlarning xotirada joylashish tartibi turli variantlarda bo'ladi. Masalan, funksiya parametrlar ro'yxatida $n \times m$ o'lchamdagi haqiqiy turdagi $x[n][m]$ massivga mos keluvchi parametrli funksiya,

```
float func(float x[n][m]);
```

ko'rinishda yozib bo'lmaydi. Masala yechimi quyidagicha bo'ladi, ya'ni bu massiv o'lchamini parametr sifatida uzatishga mo'ljallangan funksiya sarlavhasini quyidagicha yozish kerak:

```
float func(int n,int m,float x[][]);
```

Bundan tashqari ko'p o'lchamli massivlarni funksiya parametrlari sifatida ishlatishda boshqa usullardan ham foydalanish mumkin.

1. Massivning ikkinchi o'lchamini o'zgarmas ifoda (son) bilan ko'rsatish: `float func(int n,float x[][16]);`

2. Ikki o'lchamli massiv ko'rsatkichlar massivi ko'rinishida aniqlangan holatlar uchun ko'rsatkichlar massivini (matrisa satrlar adreslarini) berish orqali e'lon qilish: `float func(int ,float *p[]);`

3. Ko'rsatkichlarga ko'rsatkich ko'rinishida aniqlangan dinamik massivlarni ishlatish orqali e'lon qilish: `float func(int n,float **x);`

Endi funksiya natijasi sifatida ikki o'lchamli massivni qaytaruvchi dasturni qaraymiz. Massiv elementlarining qiymatlari tasodifiy sonlardan tashkil topgan. Tasodifiy sonlar «math.h» kutubxonasidagi `random()` funksiyasi yordamida hosil qilinadi:

```

#include <iostream.h>
#include <math.h>
int **t_matr(int n,int m)
{ int **p;
  { p=new int *[n];
    for(int i=0;i<n;i++)
      { p[i]=new int[m];
        for(int j=0;j<m;j++) p[i][j]=random(200);
      }
  }
  return p;    }

```

```

int func(int n,int m,int **ix)
{ float s=0;
  for(int i=0;i<n;i++)
    for(int j=0;j<m;j++) s+=ix[i][j];
  return s;
}

void main()
{
  int n,m;
  cin>>n>>m;
  int **matr;
  randomize();
  matr=t_matr(n,m);
  for(int i=0;i<n;i++)
  { cout<<endl<<i<<" - satr:"
    { for(int j=0;j<m;j++) cout<<"t\ "<<matr[i][j];
    }
  }
  cout<<endl<<"Summa="<<func(n,m,matr);
  for(int i=0;i<n;i++) delete matr[i];
  delete[]matr; }

```

Yuqorida keltirilgan tushunchalardan foydalanib, quyida berilgan masalalarni yechish tavsiya etiladi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 8.001** n o'lchamga ega ($n > 0$), butun sonli a massiv sonlari orasidan eng kichik sonni topuvchi butun turga tegishli $MinElem(n,a)$ nomli funksiya tasvirlansin. Bu funksiya yordamida n_a, n_b, n_c o'lchamga ega a, b, c massivlardagi eng kichik sonlar topilsin.
- 8.002** n o'lchamga ega ($n > 0$), butun sonli a massiv sonlari orasidan eng katta son joylashgan massiv indeksini topuvchi butun turga tegishli $MaxNum(n,a)$ nomli funksiya tasvirlansin. Bu funksiya yordamida n_a, n_b, n_c o'lchamga ega a, b, c massivlardagi eng katta sonlar joylashgan massiv indeksleri topilsin.
- 8.003** n o'lchamga ega ($n > 0$), haqiqiy sonli a massivning eng katta va eng kichik sonlari joylashgan massiv indekslarini topuvchi $MinmaxNum(n, a, nmin, nmax)$ nomli void turidagi funksiya tasvirlansin. $nmin$ va $nmax$ lar butun turga tegishli chiquvchi

parametrlar. Bu funksiyadan foydalanib butun turga tegishli n_a , n_b , n_c o'lchamga ega a , b , c massivlarning eng katta va eng kichik sonlari joylashgan massiv indeksleri topilsin.

- 8.004** n o'lchamga ega ($n > 0$), haqiqiy sonli a massiv sonlari ketma-ketligini teskari tartibga almashtiruvchi *Invert*(n, a) nomli void turidagi funksiya tasvirlansin. a massiv kiruvchi va chiquvchi parametr hisoblanadi. n_a , n_b , n_c o'lchamga ega a , b , c massiv sonlari bu funksiya yordamida teskari tartibda tartiblansin.
- 8.005** n o'lchamga ega ($n > 0$), haqiqiy sonli a massivni quyidagi ko'rinishda silliqlashni bajaruvchi *Smooth1*(n, a) nomli void turidagi funksiya tasvirlansin: Massivning a_k elementi dastlabki k ta elementlarning o'rta arifmetigi bilan almashtirilsin. Bu funksiya yordamida berilgan a massivda 5 karrali silliqlash bajarilsin va har bir silliqlash jarayoni chop etilsin.
- 8.006** n o'lchamga ega ($n > 0$), haqiqiy sonli a massivni quyidagi ko'rinishda silliqlashni bajaruvchi *Smooth2*(n, a) nomli void turidagi funksiya tasvirlansin: Massivdagi a_k ($k=2, \dots, n$) element a_{k-1} va a_k elementlarining yarim yig'indisi bilan almashtirilsin. Bu funksiya yordamida berilgan a massivda 5 karrali silliqlash bajarilsin va har bir silliqlash jarayoni chop etilsin.
- 8.007** n o'lchamga ega ($n > 0$), haqiqiy sonli a massivni quyidagi ko'rinishda silliqlashni bajaruvchi *Smooth3*(n, a) nomli void turidagi funksiya tasvirlansin: berilgan massivning har bir elementi, o'zining qo'shni elementlari bilan hisoblangan o'rta arifmetigiga almashtirilsin. Bu funksiya yordamida berilgan a massivda 5 karrali silliqlash bajarilsin va har bir silliqlash jarayoni chop etilsin.
- 8.008** n o'lchamga ega ($n > 0$), butun sonli a massivning x ga teng bo'lgan sonini o'chiradigan *RemoveX*(n, a, x) nomli void turidagi funksiya tasvirlansin. a massiv va n soni kiruvchi va chiquvchi parametrlar hisoblanadi. Bu funksiyadan foydalanib n_a , n_b , n_c o'lchamga ega a , b , c massivlarning x_a , x_b , x_c sonlari o'chirilsin, hosil qilingan massivlar va ularning o'lchami chiqarilsin.
- 8.009** n o'lchamga ega ($n > 0$), haqiqiy turga tegishli a massivning o'suvchi tartibda bo'lishiga halaqit beradigan elementlarini o'chiradigan *RemoveForInc*(n, a) nomli void turidagi funksiya tasvirlansin(n va a lar kiruvchi va chiquvchi parametrlar).
- 8.010** n o'lchamga ega ($n > 0$), butun sonli a massivning x ga teng bo'lgan sonlarini ikkilantiradigan *DoubleX*(n, a, x) nomli void

turidagi funksiya tasvirlansin. a massiv va n soni kiruvchi va chiquvchi parametrlar hisoblanadi. Bu funksiya foydalanib n_a , n_b , n_c o'lchamga ega a , b , c massivlarning x_a , x_b , x_c sonlari ikkilantirilsin va hosil qilingan massivlar va ularning o'lchamlari chiqarilsin.

- 8.011** n o'lchamga ega ($n > 0$), haqiqiy turga tegishli a massivni o'sish tartibida tartiblovchi *SortArray*(n, a) nomli void turidagi funksiya tasvirlansin. a massiv kiruvchi va chiquvchi parametr hisoblanadi. Bu funksiya yordamida n_a , n_b , n_c o'lchamga ega a , b , c massivlar tartiblansin.
- 8.012** n o'lchamga ega ($n > 0$), haqiqiy turga tegishli a massiv elementlarini quyidagi ko'rinishda tartiblovchi *Bell*(n, a) nomli void turidagi funksiya tasvirlansin: 1-songa hamma sonlarning eng kichigi, oxirgi songa qolgan hamma sonlarning eng kichigi, 2-songa qolgan sonlarning eng kichigi va oxiridan oldingi songa qolganlarini eng kichigi va hokazo joylashtiriladi. a massiv kiruvchi va chiquvchi parametr hisoblanadi. Bu funksiya yordamida n_a , n_b , n_c o'lchamga ega a , b , c massivlar tartiblansin.

B toifa

- 8.013** Berilgan n_a o'lchamga ega, haqiqiy turga tegishli a massivning juft o'rindagi sonlarini b massivga, toq o'rindagilarni c massivga o'tkazadigan *Split2*(n, a, b, c) nomli void turidagi funksiya tasvirlansin. Bu yerda b va c massivlar chiquvchi parametrlar. Bu funksiya n o'lchamga ega a massiv uchun qo'llanilsin va hosil bo'lgan b va c massivlar chop etilsin.
- 8.014** k o'lchamga ega, haqiqiy turga tegishli a massivdan foydalanib, $n \times m$ o'lchamga ega b matritsani quradigan (matritsa satr bo'yicha to'ldiriladi), agar a massiv sonlari ortiqcha bo'lsa, tashlab yuboradigan, kam bo'lsa 0 sonlar qo'shadigan *ArrayToMatrRow*(a, k, n, m, b) nomli void turidagi funksiya tasvirlansin. 2 o'lchovli b massiv chiquvchi parametr. Bu funksiya yordamida k o'lchamga ega a massivdan $n \times m$ o'lchamga ega b matritsa tashkil etilsin.
- 8.015** k o'lchamga ega, haqiqiy turga tegishli a massivdan foydalanib, $n \times m$ o'lchamga ega b matritsani quradigan, (matritsa ustun bo'yicha to'ldiriladi) agar a massiv sonlari ortiqcha bo'lsa, tashlab yuboradigan, kam bo'lsa 0 sonlar qo'shadigan *ArrayToMatrCol*(a, k, n, m, b) nomli void turidagi funksiya

tasvirlansin. 2 o'lchovli b massiv chiquvchi parametr. Bu funksiya yordamida k o'lchamga ega a massivdan $m \times n$ o'lchamga ega b matritsa tashkil etilsin.

8.016 Butun musbat n va m sonlari asosida $n \times m$ o'lchamga ega a matritsani tashkil etadigan, ya'ni matritsa sonlarini shaxmat doskasi ko'rinishda $(a_{0,0}=0, a_{0,1}=1, \dots)$ tasvirlaydigan $Chessboard(n, m, a)$ nomli void turidagi funksiya tuzilsin. 2 o'lchovli butun sonli a massiv chiquvchi parametr. Bu funksiya yordamida berilgan n, m butun sonlaridan foydalanib a matritsa qurilsin.

8.017 $n \times m$ o'lchamga ega haqiqiy turga tegishli a matritsaning normasini:

$$Norm1(a, n, m) = \max\left\{\sum_{j=0}^{n-1}|a_{j,0}|, \sum_{j=0}^{n-1}|a_{j,1}|, \dots, \sum_{j=0}^{n-1}|a_{j,m-1}|\right\}$$

formula asosida hisoblovchi haqiqiy turga tegishli $Norm1(a, n, m)$ nomli funksiya tasvirlansin. Bu funksiya yordamida a matritsaning normasi hisoblansin.

8.018 $n \times m$ o'lchamga ega haqiqiy turga tegishli a matritsaning normasini: $Norm2(a, n, m) = \max\left\{\sum_{j=0}^{m-1}|a_{0,j}|, \sum_{j=0}^{m-1}|a_{1,j}|, \dots, \sum_{j=0}^{m-1}|a_{n-1,j}|\right\}$ formula

asosida hisoblovchi haqiqiy turga tegishli $Norm2(a, n, m)$ nomli funksiya tasvirlansin. Bu funksiya yordamida a matritsaning normasi hisoblansin.

8.019 $n \times m$ o'lchamga ega haqiqiy sonli a matritsaning k -satridagi sonlari yig'indisini hisoblovchi (agar $k > n$ bo'lsa 0 qiymat qaytaradigan) haqiqiy turga tegishli $SumRow(n, m, a, k)$ nomli funksiya tasvirlansin. Bu funksiya yordamida a matritsa berilganda, k ning 3 ta turli qiymatlarida funksiyaning qaytaradigan natijalari chop etilsin.

8.020 $n \times m$ o'lchamga ega haqiqiy sonli a matritsaning k -ustunidagi sonlari yig'indisini hisoblovchi (Agar $k > m-1$ bo'lsa 0 qiymat qaytaradigan) haqiqiy turga tegishli $SumCol(n, m, a, k)$ nomli funksiya tasvirlansin. Bu funksiya yordamida a matritsa berilganda k ning 3 ta turli qiymatlarida funksiyaning qaytaradigan natijalari chop etilsin.

8.021 $n \times m$ o'lchamga ega haqiqiy sonli a matritsaning k_1 va k_2 satrlari o'rinlarini o'zaro almashtirishni bajaruvchi, agar $k_1 > n-1$ yoki $k_2 > n-1$ bo'lsa, matritsani o'zgarishsiz qoldiruvchi $SwapRow(n, m, a, k_1, k_2)$ nomli void turidagi funksiya tasvirlansin. a

matritsa kiruvchi va chiquvchi parametr hisoblanadi. Bu funksiyadan foydalanib a matritsaning k_1 va k_2 satrlarining o'rinlari o'zaro almashtirilsin.

- 8.022** $n \times m$ o'lchamga ega haqiqiy sonli a matritsaning k_1 va k_2 ustunlari o'rinlarini o'zaro almashtirishni bajaruvchi, agar $k_1 > m-1$ yoki $k_2 > m-1$ bo'lsa, matritsani o'zgarishsiz qoldiruvchi **Swapcol**(n, m, a, k_1, k_2) nomli void turidagi funksiya tasvirlansin. a matritsa kiruvchi va chiquvchi parametr hisoblanadi. Bu funksiyadan foydalanib, a matritsaning k_1 va k_2 satrlarining o'rinlari o'zaro almashtirilsin.
- 8.023** n tartibli haqiqiy sonli a kvadrat matritsani transponerlashni bajaradigan **Transp**(n, a) nomli void turidagi funksiya tasvirlansin. a kiruvchi va chiquvchi parametr. Bu funksiya yordamida berilgan n tartibli a matritsa transponerlansin.
- 8.024** $n \times m$ o'lchamga ega haqiqiy sonli a matritsaning k_1 va k_2 hamda ular orasidagi satrlarni ($k_1 < k_2$) o'chiruvchi **RemoveRows**(n, m, a, k_1, k_2) nomli void turidagi funksiya tasvirlansin. n va a kiruvchi va chiquvchi parametrlar. Agar $k_1 > n-1$ bo'lsa, matritsa o'zgarishsiz qoldirilsin. Agar $k_2 > n-1$ bo'lsa, k_1 dan $n-1$ gacha bo'lgan barcha satrlar o'chirilsin.
- 8.025** $n \times m$ o'lchamga ega haqiqiy sonli a matritsaning k_1 va k_2 hamda ular orasidagi ustunlarni ($k_1 < k_2$) o'chiruvchi **RemoveCols**(n, m, a, k_1, k_2) nomli void turidagi funksiya tasvirlansin. n va a kiruvchi va chiquvchi parametrlar. Agar $k_1 > m-1$ bo'lsa, matritsa o'zgarishsiz qoldirilsin. Agar $k_2 > m-1$ bo'lsa k_1 dan $m-1$ gacha bo'lgan barcha ustunlari o'chirilsin.
- 8.026** $n \times m$ o'lchamga ega a matritsaning har bir ustunini o'sish tartibida tartiblaydigan **SortCols**(n, m, a) nomli void turidagi funksiya tasvirlansin. Ikki o'lchovli a massiv kiruvchi va chiquvchi parametr hisoblanadi. Funksiyadan foydalanib berilgan a matritsa ustunlari tartiblansin.

8.2. Satrlar va fayllarga doir masalalarni yechishda funksiyalardan foydalanish

Satrlar bilan ishlashda bir-biriga o'xshash bo'lgan jarayonlar ko'plab uchraydi. Bunday holatlarda dasturchi tomonidan kiritiladigan funksiyalardan foydalanish qo'yilgan masalalarni yechishda samarali usullardan biri hisoblanadi.

Fayllar bilan ishlashga bog'liq masalalarda ham yuqoridagi kabi mulohazalarni keltirish mumkin. Ayniqsa fayllarni ochish, yopish jarayonlari bitta dasturda bir necha marotaba takrorlanadi, shuning uchun ham bu tipdagi masalalarda dasturchi tomonidan funksiyalardan keng foydalanish muhim ahamiyatga ega. Funksiya parametri sifatida satr turiga tegishli o'zgaruvchilarni uzatish (fayl nomlarini ham) imkoniyatlari ham mavjud. Quyida keltirilgan misolda bu mulohazalar o'z aksini topgan.

Masala. Fayl nomi va satr berilgan bo'lsa, berilgan nomdagi faylni tuzish va unga ko'rsatilgan satrni yozishni amalga oshiruvchi *f_ochish(s1,s2)* nomli void turidagi funksiya tuzilsin. Bu yerda *s1* va *s2* lar kiruvchi parametrlar bo'lib, *char* turidagi belgilar massivi. Bu funksiyadan foydalanib, 3 ta fayl yaratilsin va ularning har biriga berilgan satr yozilsin.

```
#include <cstdlib>
#include <iostream>
using namespace std;
/* char turi bazasida 30 tagacha belgi qabul qila oladigan "sotr" nomli
tur e'lon qilish*/
typedef char sotr[30];
void f_ochish(sotr s1,sotr s2);
int main(int argc, char *argv[])
{
    sotr s,s0="Salom talabalar";
    for (int i=1; i<=3; i++)
    {
        cout<<i<<" -fayl nomini kiriting";
        cin>>s;
        f_ochish(s,s0);
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
void f_ochish(sotr s1,sotr s2)
{
    FILE *f1; // fayl turidagi o'zgaruvchini e'lon qilish;
    // f1 o'zgaruvchiga bog'langan faylni ochish;
    if((f1=fopen(s1,"wt+"))==NULL)
        cout<<s1<<"fayli ochilmadi!!\n";
```

```

    fputs(s2,f1); /* s2 o'zgaruvchisidagi ma'lumotni f1 o'zgaruvchisi bilan
    bog'langan faylga yozish; */
    // f1 o'zgaruvchisi bilan bog'langan faylni yopish;
    fclose(f1); }

```

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 8.027** Berilgan s satrning ruxsat etilgan identifikator ekanligini tekshiruvchi va s bo'sh bo'lmagan identifikator bo'lsa 0, s bo'sh satr bo'lsa 1, s raqam bilan boshlansa 2, s da ruxsat etilmagan belgilar bo'lsa, uning 1-uchragan ruxsat etilmagan belgisining kodini qaytaradigan butun turga tegishli *Isident(s)* nomli funksiya tasvirlansin. s satr turidagi parametr. 3 ta har xil berilgan satrlar uchun *Isident* funksiyaning qaytaradigan qiymatlari chop etilsin.
- 8.028** s satr shablonini takror yozish natijasida hosil qilingan n uzunlikdagi satrni qaytaruvchi satr turga tegishli *FillStr(s,n)* nomli funksiya tasvirlansin. s satr, n butun turdagi parametrlar. Bu funksiya yordamida berilgan n soni va 3 ta satr uchun har biri n uzunlikka teng bo'lgan 3 ta satr shablони chop etilsin.
- 8.029** Lotin kichik harflaridan iborat s satrdagi harflarni katta lotin harflariga o'tkazadigan (belgilar o'zgartirilmaydi) *UpCaseLat(s)* nomli void turidagi funksiya tasvirlansin. s satr kiruvchi va chiquvchi parametr. Funksiyadan foydalanib 3 ta satrning harflari katta harflarga o'tkazilsin.
- 8.030** Lotin katta harflaridan iborat s satrdagi harflarni kichik lotin harflariga o'tkazadigan (belgilar o'zgartirilmaydi) *LowCaseLat(s)* nomli void turidagi funksiya tasvirlansin. s satr kiruvchi va chiquvchi parametr. Funksiyadan foydalanib 5 ta satrning harflari katta harflarga o'tkazilsin.
- 8.031** s satrning c belgi bilan ustma-ust tushadigan 1-belgisini o'chiruvchi *TrimLeftC(s,c)* nomli void turidagi funksiya tasvirlansin. s satr va c belgi turidagi parametrlar. Ixtiyoriy belgi va 3 ta satr berilgan. Funksiyadan foydalanib satrlarning yangi ko'rinishi tasvirlansin.
- 8.032** s satrning oxirgi belgisi bilan ustma-ust tushadigan belgilarini o'chiruvchi *TrimRight(s,c)* nomli void turidagi funksiya tasvirlansin. s satr kiruvchi va chiquvchi parametr. 3 ta satr

berilganda funksiyadan foydalanib satrlarning yangi ko‘rinishi tasvirlansin.

- 8.033** s satrdagi k -pozitsiyadan boshlanuvchi n ta belgidan iborat qism satrni teskari tartibda tartiblovchi satr turga tegishli $InvertStr(s,k,n)$ nomli funksiya tasvirlansin. Agar k satr uzunligidan katta bo‘lsa bo‘sh joy, agar satr uzunligi $k+n$ dan kichik bo‘lsa, berilgan satrning barchasini teskari tartiblashni funksiya bajarishi kerak. Berilgan satr va 3 ta musbat (k_1, n_1) , (k_2, n_2) , (k_3, n_3) juftliklar uchun funksiyaning qiymati chiqarilsin.
- 8.034** s satrdan oxirgi topilgan s_0 qism satrning boshlang‘ich pozitsiyasini qaytaruvchi butun turga tegishli $PosLast(s, s_0)$ nomli funksiya tasvirlansin. Agar satrda qism satr topilmasa funksiya 0 qiymat qaytarsin. Bu funksiyadan foydalanib s satr uchun 3 ta s_0 qism satrlarga mos keluvchi qiymatlar topilsin.
- 8.035** s satrdagi 1-topilgan s_0 qism satrning boshlang‘ich pozitsiyasini qaytaruvchi butun turga tegishli $PosK(s, s_0)$ nomli funksiya tasvirlansin. Agar satrda qism satr topilmasa funksiya 0 qiymat qaytarsin. Berilgan s satr uchun 3 ta s_0 qism satrlarning joylashuv o‘rni hisoblansin.
- 8.036** s satrdagi k -so‘zni qaytaruvchi satr turga tegishli $WordK(s, k)$ nomli funksiya tasvirlansin. Agar satrdagi so‘zlar soni k dan kam bo‘lsa, funksiya bo‘sh satr qaytarsin. Bu funksiya yordamida berilgan satrdan ajratilgan k_1 -, k_2 -, k_3 -so‘zlar chop etilsin.
- 8.037** s satr so‘zlaridan w massivni tuzadigan $SplitStr(s, w, n)$ nomli void turidagi funksiya tasvirlansin (w massiv va uning o‘lchami n chiquvchi parametr). Funksiyadan foydalanib berilgan s satrdagi so‘zlar soni va ularning o‘zlari aniqlansin.
- 8.038** s satrning 4 tadan kam bo‘lmagan ketma-ket keluvchi bir xil belgilarini $s\{n\}$ qisqartma bilan almashtiradigan zichlovchi satr turga tegishli $CompressStr(s)$ nomli funksiya tasvirlansin. Bu funksiya yordamida berilgan 5 ta satr zichlashtirilsin. Misol. $S = "bbccccce"$ satr uchun funksiya $bbbc\{5\}e$
- 8.039** $CompressStr$ nomli funksiya yordamida zichlashtirilgan s satrni asl holiga qaytaradigan satr turga tegishli $DecompressStr(s)$ nomli funksiya tasvirlansin. Bu funksiya yordamida zichlashtirilgan 5 ta satr asl holiga qaytarilsin.
- 8.040** O‘nlik sanoq sistemasidagi n butun musbat sonining ikkilik sanoq sistemasidagi tasvirini satr ko‘rinishda qaytaradigan satr turga tegishli $DecToBin(n)$ nomli funksiya tasvirlansin. Berilgan

3 ta son uchun bu funksiyadan foydalanib ularning ikkilik tasvirlari hosil qilinsin.

- 8.041** O'nlik sanoq sistemasidagi n butun musbat sonining o'n oltilik sanoq sistemasidagi tasvirini satr ko'rinishda qaytaradigan satr turga tegishli *DecToNex(n)* nomli funksiya tasvirlansin. Berilgan 3 ta son uchun bu funksiyadan foydalanib ularning o'n oltilik tasvirlari hosil qilinsin.
- 8.042** s satrda ikkilik sanoq sistemasidagi son joylashgan bo'lsa, satrdagi qiymatga mos keluvchi butun musbat o'nlik sonni qaytaruvchi butun turga tegishli *BinToDec(s)* nomli funksiya tasvirlansin. Bu funksiya yordamida satr ko'rinishidagi 3 ta ikkilik sonlar o'nlik sonlarga o'tkazilsin.
- 8.043** s satrda o'n oltilik sanoq sistemasidagi son joylashgan bo'lsa, satrdagi qiymatga mos keluvchi butun musbat o'nlik sonni qaytaruvchi butun turga tegishli *NexToDec(s)* nomli funksiya tasvirlansin. Bu funksiya yordamida satr ko'rinishidagi 3 ta ikkilik sonlar o'nlik sonlarga o'tkazilsin.

B toifa

- 8.044** s nomli fayldagi belgilar sonini qaytaradigan butun turga tegishli *IntFileSize(s)* nomli funksiya tasvirlansin. Agar fayl mavjud bo'lmasa funksiya -1 qiymatni qaytarsin. Bu funksiya yordamida berilgan nomdagi 3 ta fayldagi belgilar soni aniqlansin.
- 8.045** s matnli fayldagi satrlar sonini qaytaruvchi, butun turga tegishli *LineCount(s)* nomli funksiya tasvirlansin. Bu funksiya yordamida 3 ta matnli fayllarning har biri uchun satrlar soni aniqlansin. Agar ko'rsatilgan nomdagi matnli fayl mavjud bo'lmasa funksiya -1 ni qaytarsin.
- 8.046** Butun sonlarni saqlaydigan s fayldgi sonlarning joylashish tartibini teskarisiga almashtiradigan *InvertIntFile(s)* nomli void turidagi funksiya tasvirlansin. 3 ta fayldagi ma'lumotlar shu funksiya yordamida qayta ishlansin. Agar fayl mavjud bo'lmasa yoki fayldagi sonlar miqdori 2 tadan kam bo'lsa, funksiya hech qanday amal bajarmasin.
- 8.047** s matnli fayldagi har bir satrining boshiga nomer qo'yadigan *AddLineNumbers(s, n, k)* nomli void turidagi funksiya tasvirlansin, satrlar tartib nomeri quyidagicha: 1-satr n , 2-si $n+1$ va hokazo. Nomerlash k -chi pozitsiyadan boshlansin ($k > 0$, n , k lardan foydalangan holda ushbu funksiyani bo'sh bo'lmagan fayl

uchun qo'llang).

- 8.048** *AddLineNumbers* nomli void turidagi funksiy ayordamida har bir satri nomerlangan s matn turga tegishli fayldagi satr nomerlarini o'chiradigan *RemoveLineNumbers(s)* nomli void turidagi funksiya tasvirlansin. Bu funksiya berilgan nomdagi fayl uchun qo'llanilsin. Agar satrlar nomerlanmagan bo'lsa, funksiya hech qanday harakat bajarmasin.
- 8.049** Butun sonlardan iborat s fayldagi k ta ($k \geq 0$) sonni s_1 faylga, qolgan qismini esa s_2 faylga o'tkazuvchi *SplitintFile(s₀,k,s₁,s₂)* nomli void turidagi funksiya tasvirlansin. Bu erda s_1 va s_2 fayllardan birortasi bo'sh qolishi ham mumkin. Ko'rsatilgan k , s_1 va s_2 qiymatlarda berilgan fayl uchun ushbu funksiya qo'llanilsin.
- 8.050** s_0 matn turga tegishli fayldagi k ta ($k \geq 0$) satrni s_1 faylga, qolgan qismini s_2 fayliga o'tkazuvchi *SplitText(s₀,k,s₁,s₂)* nomli void turidagi funksiya tasvirlansin. Ko'rsatilgan k , s_1 va s_2 qiymatlarda berilgan fayl uchun ushbu funksiya qo'llanilsin.
- 8.051** 2 lik satrlarga ega fayldagi ma'lumotlarni matn ko'rinishiga aylantiruvchi *StringFileToText(s)* nomli void turidagi funksiya tasvirlansin. Bu funksiya dan foydalanib, berilgan 2 ta 2 lik satr fayllaridagi ma'lumotlar matn ko'rinishiga o'tkazilsin.
- 8.052** s matnli fayldagi ma'lumotlarni 2 lik satr ko'rinishiga o'tkazuvchi *TextToStringFile(s)* nomli void turidagi funksiya tasvirlansin. Ushbu funksiya dan foydalanib, 2 ta matn turga tegishli fayllardagi ma'lumotlar 2 lik satr ko'rinishiga o'tkazilsin.
- 8.053** s matnli fayldagi ma'lumotni shifrlaydigan *EncodeText(s,k)* nomli void turidagi funksiya tasvirlansin. Shefrlash alfavitdagi harf nomeri asosida amalga oshirilsin, siljish $k(0 < k < 10)$ qadam bo'lsin. Misol uchun: $k=3$ bo'lsa "a" o'rniga "d" harfi. Alfavitga kirmaydigan belgilar o'zgarishsiz qoldirilsin. Ushbu funksiya dan foydalanib berilgan nomdagi fayl k kalit bo'yicha shifrlansin.
- 8.054** s matnli fayldagi k kalit bilan shifrlangan ma'lumotlarni o'z holiga qaytaruvchi *DecodeText(s,k)* nomli void turidagi funksiya tasvirlansin. Ushbu funksiya dan foydalanib, berilgan nomdagi matnli faylga joylashgan, k kalit bo'yicha shifrlangan ma'lumotlar asl holiga qaytarilsin.

8.3. Strukturalar

Amaliyotda uchraydigan biror sohaga tegishli masalalarni yechishda unga tegishli ob'ektlar bir necha turdaga parametrlar bilan aniqlanishi mumkin. Masalan, talaba haqidagi ma'lumotlar: satr turidagi talaba familiyasi, ismi va sharifi, (mutaxassislik) yo'nalishi, uning yashash manzili;

Butun turdagi tug'ilgan yili, kursi, guruhi, o'qiydigan fanlar soni;

Haqiqiy turdagi reyting bali, mantiqiy turdagi talaba jinsi kabilardan shakllanadi.

Dasturda holat yoki tushunchani tavsiflovchi har bir berilganlar uchun alohida-alohida o'zgaruvchilarni aniqlab, qo'yilgan masalani yechish mumkin. Ammo bu holda obyekt haqidagi ma'lumotlar «tarqoq» bo'ladi, ularni qayta ishlash murakkablashadi, ya'ni obyekt haqidagi berilganlarni yaxlit holda ko'rish qiyinlashadi.

C++ tilida bir yoki bir necha turdagi berilganlarni bir-biriga bog'lash imkoniyati mavjud bo'lib, bunday bog'lanish *struktura* deb ataladi. Struktura foydalanuvchi tomonidan aniqlangan berilganlarning yangi turi hisoblanadi va u quyidagicha aniqlanadi:

```
struct <struktura nomi>
{
    <tur_1> <o'zgaruvchi_1>;
    <tur_2> <o'zgaruvchi_2>;
    ...
    <tur_n> <o'zgaruvchi_n>;
};
```

Bu yerda <struktura nomi> – struktura ko'rinishida yaratilayotgan yangi turning nomi, <tur_i> <o'zgaruvchi_i>;(i=1..n) – strukturaning i-o'zgaruvchisining e'loni.

Boshqacha qilib aytganda, struktura–e'lon qilingan o'zgaruvchilardan (maydonlardan) tashkil topadi. Uni har xil turdagi berilganlarni o'z ichiga oluvchi qobiq deb qarash mumkin. Qobiqdagi berilganlarni yaxlit holda ko'chirish, tashqi qurilmalar (binar fayllar)ga yozish, o'qish mumkin bo'ladi.

Talaba haqidagi berilganlarni o'z ichiga oluvchi struktura turining e'lon qilinishini ko'raylik.

```
struct Talaba
{
```

```
char FISH[30];
unsigned int Tug_yil;
unsigned int Kurs,Uk_fan_soni;
char Yunalish[30];
float Reyting;
bool Jinsi;
char Manzil[60];
};
```

Struktura turining e'lonida *<tur nomi>* bo'lmasligi ham mumkin, bu holda struktura aniqlangandan keyin albatta shu struktura turi bilan aniqlanadigan o'zgaruvchilarning nomlari yozilishi kerak. Masalan,

```
struct
{
    char FISH[30];
    unsigned int Tug_yil;
    unsigned int Kurs,Uk_fan_soni;
    char Yunalish[30];
    float Reyting;
    bool Jinsi;
    char Manzil[60];
} Talaba1,Talaba2;
```

Keltirilgan misolda struktura turidagi *Talaba1*, *Talaba2* o'zgaruvchilari e'lon qilingan.

Struktura turidagi o'zgaruvchilar bilan ishlash, uning maydonlari bilan ishlashni anglatadi. Struktura maydoniga murojaat qilish '.' (nuqta) orqali amalga oshiriladi. Bunda struktura turidagi o'zgaruvchi nomi, undan keyin nuqta qo'yiladi va maydon o'zgaruvchisining nomi yoziladi. Masalan, talaba haqidagi struktura maydonlariga murojaat quyidagicha bo'ladi:

```
talaba.Kurs=1;
talaba.Tug_yil=1991;
strcpy(talaba.FISH,"Eshbutayev E.A.");
strcpy(talaba.Yunalish,
    "Amaliy matematika va informatika");
strcpy(talaba.Manzil,
    "Qarshi sh,Koçhabog 17,tel: 225-43-25");
talaba.Reyting=72.8;
```

Keltirilgan misolda talaba strukturasining son turidagi maydonlariga oddiy ko'rinishda qiymatlar berilgan, satr turidagi maydonlar uchun *strcpy* funksiyasi orqali qiymat berish amalga oshirilgan.

Struktura turidagi obyektning xotiradan qancha joy egallaganligi *sizeof* funksiyasi yordamida aniqlanadi:

```
int i=sizeof(Talaba);
```

Ayrim hollarda struktura maydonlari o'lchamini bitlarda aniqlash orqali egallanadigan xotirani kamaytirish mumkin. Buning uchun struktura maydoni quyidagicha e'lon qilinishi kerak:

```
<maydon nomi> : <o'zgarmas ifoda>;
```

Bu erda *<maydon nomi>* – maydon turi va nomi, *<o'zgarmas ifoda>*—maydonning bitlardagi uzunligi. Bunday elondagi maydon turi butun turga tegishli bo'lishi kerak (*int, long, unsigned, char*).

Agar foydalanuvchi strukturaning maydoni faqat 0 va 1 qiymatini qabul qilishini bilsa, bu maydon uchun bir bit joy ajratishi mumkin (bir bayt yoki ikki bayt o'rniga). Xotirani tejash evaziga maydon ustida amal bajarishda razryadli arifmetikani qo'llash zarur bo'ladi.

Misol uchun sana–vaqt bilan bog'liq strukturani yaratishni qaraylik. Struktura yil, oy, kun, soat, minut va sekund maydonlaridan iborat bo'lsa, unga mos turni quyidagicha e'lon qilish mumkin:

```
struct Sana_vaqt
{ unsigned Yil : 7;
  unsigned Oy: 4;
  unsigned Kun: 5;
  unsigned Soat: 5;
  unsigned Minut: 6;
  unsigned Sekund: 6; };
```

bu struktura xotiradan 5 baytga yaqin joy egallaydi.

Strukturalar funksiya argumenti sifatida ishlatilishi mumkin. Buning uchun funksiya prototipida struktura turi ko'rsatilishi kerak bo'ladi. Masalan, talaba haqidagi berilganlarni o'z ichiga oluvchi Talaba strukturasini turidagi berilganlarni *Talaba_Manzili()* funksiyasiga parametr sifatida berish uchun funksiya prototipi quyidagi ko'rinishda bo'lishi kerak:

```
void Talaba_Manzili(Talaba);
```

Funksiyaga struktura turidagi o'zgaruvchini argument sifatida uzatishni quyidagi dasturda namoyish etamiz:

```
#include <iostream.h>
#include <string.h>
```

```
struct Talaba
{
    char FISH[30];
    unsigned int Tug_yil;
    unsigned int Kurs;
    char Yunalish[30];
    float Reyting;
    char Manzil[60];
    bool jinsi;
};
void Talaba_Manzili(Talaba);
int main()
{
    Talaba talaba;
    talaba.Kurs=1;
    talaba.tug_yil=1991;
    strcpy(talaba.FISH, "Eshbutayev E.A.");
    strcpy(talaba.Yunalish,
        "Amaliy matematika va informatika");
    strcpy(talaba.Manzil,
        "Qarshi sh,Kochabog 17,tel: 225-43-25");
    talaba.Reyting=72.8;
    Talaba_Manzili(talaba);
    return 0;
}
void Talaba_Manzili(Talaba t)
{
    cout<<"Talaba FIO: "<<t.FISH<<endl;
    cout<<"Manzili: "<<t.Manzil<<endl;
}
```

Dasturning bosh funksiyasida talaba strukturasi aniqlanib, uning maydonlariga qiymatlar berilgan. So'ngra talaba strukturasi *Talaba_Manzili()* funksiyasiga argument sifatida uzatilgan. Dastur ishlashi natijasida ekranga quyidagi ma'lumotlar chop etiladi.

Talaba FIO: *Eshbutayev E.A.*

Manzili: *Qarshi sh,Kochabog 17,tel: 225-43-25.*

O'z-o'zidan ma'lumki, struktura turidagi yagona berilgan bilan yechish mumkin bo'lgan masalalar doirasi juda tor va aksariyat holatlarda, qo'yilgan masala strukturalar majmuasi bilan ishlashni talab

qiladi. Bu turdagi masalalarga berilganlar bazasini qayta ishlash masalalari deb qarash mumkin.

Strukturalar massivini e'lon qilish xuddi standart massivlarni e'lon qilishdek bo'lib, uning farqi massiv turi o'rnida foydalanuvchi tomonidan aniqlangan struktura turi yoziladi. Masalan, talabalar xaqidagi berilganlarni o'z ichiga olgan massiv yaratish e'loni quyidagicha bo'ladi:

```
const int n=25;
```

```
Talaba talabalar[n];
```

Strukturalar massivining elementlariga murojaat odatdagi massiv elementlariga murojaat usullari orqali, har bir elementning maydonlariga murojaat esa '.' orqali amalga oshiriladi, ya'ni *talabalar[1].kurs=3;* kabi.

Struktura elementlariga ko'rsatkichlar orqali ham murojaat qilish mumkin. Buning uchun strukturaga ko'rsatkich o'zgaruvchisi e'lon qilinishi kerak. Masalan, yuqorida keltirilgan misolda "Talaba" strukturasi ko'rsatkich e'lon qilinsa, u

```
Talaba *k_talaba;
```

ko'rinishida bo'ladi.

Ko'rsatkich orqali aniqlangan struktura elementlariga murojaat «.» bilan emas, balki «->» vositasida amalga oshiriladi, ya'ni

```
cout<<k_talaba->FISH; kabi.
```

Berilganlar bilan ishlashda ularning miqdori qancha bo'lishi va ularga xotiradan qancha joy ajratilishi ko'p hollarda oldindan ma'lum bo'lmaydi. Dastur ishlash paytida berilganlar uchun zarurat bo'yicha xotiradan joy ajratish va ularni ko'rsatkichlar bilan bog'lash orqali yagona struktura hosil qilish jarayoni xotiraning dinamik taqsimoti deyiladi. Bu usulda hosil bo'lgan berilganlar majmuasiga berilganlarning dinamik strukturasi deyiladi, chunki ularning o'lchami dastur bajarilishida o'zgarib turadi. Dasturlashda dinamik strukturalardan foydalanib, chiziqli ro'yxatlar (zanjirlar), styoklar, navbatlar va binar daraxtlari quriladi. Ular bir-biridan elementlar o'rtasidagi bog'lanishlari va ular ustida bajariladigan amallari bilan farqlanadi. Dastur ishlashida strukturaga yangi elementlar qo'shilishi yoki o'chirilishi mumkin.

Har qanday berilganlarning dinamik strukturasi maydonlardan tashkil topadi va ularning ayrimlari qo'shni elementlar bilan bog'lanish uchun xizmat qiladi.

Quyidagi masalalar uchun dasturlar tuzilsin:

A toifa

- 8.055** Maydonlari butun turga tegishli *day(kun)*, *month(oy)* va *year(yil)* bo'lgan *tdate* struktura turi va parametri *tdate* turga tegishli bo'lgan, kabisa yil sanasi uchun 1 (*rost*), boshqa hollarda 0 (*yolg'on*) qiymat qaytaruvchi mantiqiy *Leapyear(d)* nomli funksiya tasvirlansin. Ushbu tur va funksiya foydalanib kiritilgan 5 ta sana qiymati uchun natijalar chiqarilsin.
- 8.056** *tdate* turi va *LeapYear()* nomli funksiya foydalangan holda parametri *tdate* bo'lgan, ko'rsatilgan sanaga mos keladigan oyning kunlari sonini qaytaradigan butun turga tegishli *DaysInMonth(d)* funksiya tasvirlansin. Tasvirlangan funksiya foydalanib berilgan 5 ta sananing har biriga mos oy kunlari hisoblansin.
- 8.057** *tdate* struktura turidan va *DaysInMonth()* funksiyalaridan foydalanib, parametri *tdate* turga tegishli bo'lgan, *d* sana to'g'ri bo'lsa 0; *d* sanadagi oy nomeri xato bo'lsa 1; *d* sanadagi berilgan oy uchun keltirilgan kun xato bo'lsa 2 qiymatlardan birini qaytaruvchi butun turga tegishli *CheckDate(d)* nomli funksiya tasvirlansin. Berilgan 5 ta sananing har biri uchun *CheckDate()* nomli funksiyaning qaytargan qiymatlari chiqarilsin.
- 8.058** *tdate* struktura turi, *DaysInMonth()* va *CheckDate()* funksiyalaridan foydalanib, parametri *tdate* turga tegishli bo'lgan berilgan *d* sanani o'zidan 1 kun oldingi sana bilan almashtiruvchi, agar sana xato bo'lsa o'zgarishsiz qoldiradigan, *PrevDate(d)* nomli void turidagi funksiya tasvirlansin. (Bu yerda *d* struktura turga tegishli parametr, ma'lumotlarni qabul qiladi va qaytaradi.) Bu funksiya 5 ta turli sanalar uchun qo'llanilsin.
- 8.059** *tdate* struktura turi, *DaysInMonth()* va *CheckDate()* funksiyalaridan foydalanib, parametri *tdate* turga tegishli bo'lgan berilgan *d* sanani o'zidan 1 kun keyingi sana bilan almashtiruvchi, agar sana xato bo'lsa o'zgarishsiz qoldiradigan, *NextDate(d)* nomli void turidagi funksiya tasvirlansin. (Bu yerda *d* struktura turga tegishli parametr, ma'lumotlarni qabul qiladi va qaytaradi.) Bu funksiya 5 ta turli sanalar uchun qo'llanilsin.

B toifa

- 8.060** Haqiqiy turga tegishli x va y maydonlarga ega bo'lgan *TPoint* struktura turi, hamda parametrlari *TPoint* bo'lgan kesma uzunligini hisoblovchi haqiqiy turga tegishli *Leng(a,b)* nomli funksiya tasvirlansin. ab , ac , ad kesmalar koordinatalari bilan berilgan bo'lsa, tasvirlangan funksiya dan foydalanib kesmalarning uzunliklari topilsin.
- 8.061** *TPoint* va *Leng()* funksiyalaridan foydalanib, turi *tpoint* bo'lgan a , b , c maydonlarga ega *ttriangle* struktura turi hamda haqiqiy turga tegishli parametri *ttriangle* turga tegishli bo'lgan uchburchakning perimetrini hisoblovchi haqiqiy turga tegishli *Perim(t)* nomli funksiya tasvirlansin. Agar a, b, c, d lar berilgan nuqtalar koordinatalari bo'lsa, (tekislikda) tasvirlangan funksiya dan foydalanib, abc , abd , acd uchburchaklarning perimetrlari hisoblansin.
- 8.062** *Tpoint*, *ttriangle* turlaridan va *Leng()*, *Perim()* funksiyalaridan foydalanib, parametri *ttriangle* turga tegishli bo'lgan, uchburchakning yuzini hisoblaydigan haqiqiy turga tegishli *Area(t)* nomli funksiya tasvirlansin. Agar a , b , c , d lar tekislikda berilgan nuqtalarning koordinatalari bo'lsa, tasvirlangan funksiya dan foydalanib, abc , abd , acd uchburchaklarning yuzalari hisoblansin.
- 8.063** *Tpoint*, *ttriangle* turlari va *Leng()*, *Area()* funksiyalaridan foydalanib, parametrlari *tpoint* turga tegishli bo'lgan p nuqtadan kesmagacha bo'lgan masofani hisoblaydigan haqiqiy turga tegishli *Dist(p,a,b)* nomli funksiya tasvirlansin. Agar p , a , b , c nuqtalar tekislikda koordinatalari bilan berilgan bo'lsa, tasvirlangan funksiya dan foydalanib, p nuqtadan ab , ac , bc kesmalar yotuvchi to'g'ri chiziqlargacha bo'lgan masofalar hisoblansin.
- 8.064** *Tpoint*, *ttriangle* turlari va *Dist()* nomli funksiya dan foydalanib, parametri t *ttriangle* turga tegishli bo'lgan, uchburchakning h_1 , h_2 , h_3 balandliklarini hisoblaydigan *Heights(t, h₁, h₂, h₃)* nomli void turidagi funksiya tasvirlansin. Bu yerda t qiymat qabul qiluvchi parametr, h_1 , h_2 , h_3 lar esa qiymat qaytaruvchi parametrlar hisoblanadi. Tekislikda A , B , C , D nuqtalar koordinatalari bilan berilgan bo'lsa, tasvirlangan funksiya dan foydalanib, $t.a$, $t.b$, $t.c$, $t.d$ tomonlariga tushirilgan ABC , ABD , ACD uchburchaklarning balandliklari hisoblansin.

- 8.065** *Tpoint* turi va *Leng()* nomli funksiyadan foydalanib, n burchakli ko'pburchakning perimetrini hisoblovchi, parametri *Tpoint* massiv turga tegishli bo'lgan, haqiqiy turga tegishli *PerimN(p,n)* nomli funksiya tasvirlansin. Ko'pburchakning tomonlari soni va uchlarining koordinatalari berilgan bo'lsa, tasvirlangan funksiyadan foydalanib, 3 ta berilgan ko'pburchaklarning perimetrlari hisoblansin.
- 8.066** *Tpoint*, *ttriangle* turlari va *Area()* nomli funksiyadan foydalanib, qavariq n burchakning yuzini hisoblaydigan (n -tomonlar soni, p -*tpoint* turga tegishli massiv) haqiqiy turga tegishli *AreaN(p,n)* nomli funksiya tasvirlansin. Ko'pburchaklarning tomonlari soni va uchlarining koordinatalari berilgan bo'lsa, tasvirlangan funksiyadan foydalanib, 3 ta ko'pburchaklarning har biri uchun yuzalar hisoblansin.

8.4. Bobga doir qiyinroq masalalar

8.067

G'iyos va $N!$ ning bo'luvchilari [12]

G'iyos juda tezkor dasturchi. U masalalarni qisqa vaqtda yechadi. U yaqinda *tuit.uz* saytidagi $N!$ ning bo'luvchilari masalasini yechdi. Masala sharti bo'yicha: $N!$ deb 1 dan N gacha bo'lgan natural sonlarning ko'paytmasiga aytiladi ($N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$). Sizning vazifangiz $N!$ ning natural bo'luvchilari sonini topishdan iborat. Javob yetarlicha katta bo'lishi mumkin, shuning uchun sizdan faqat uni 1000000007 ($10^9 + 7$) gacha bo'lgandagi qoldiqni topish so'raladi ($1 \leq N \leq 10^5$). Bu masalani yechish uchun G'iyos 1 hafta vaqt sarfladi. Endi uni agar $1 \leq N \leq 3 \cdot 10^6$ bo'lsa qanday yechish mumkinligi qiziqtirib qoldi. Tushinib yetdiki, bu masalani yechish uchun endi bir oy vaqt kerak bo'ladi. Unga bu masalani 4 soat ichida yechishga yordam bering.

Berilganlar:

N natural soni berilgan ($1 \leq N \leq 3 \cdot 10^6$).

Natija:

$N!$ ning natural bo'luvchilari soni miqdorining $10^9 + 7$ bo'lgan holdagi qoldig'ini chiqaring.

N	Berilganlar:	Natija:
1	1	1

2	3	4
3	20	41040

8.068

Azamat informatika bo'yicha juda ko'plab musobaqalarga qatnashadi va u qatnashgan har bir i -musobaqasini o'lchamlari w_i va h_i (w_i —eni, h_i —bo'yi) bo'lgan varaqlarga yozib boradi. Bir kuni u musobaqa natijalari yozilgan barcha varaqlarni stol ustiga taxlashga qaror qildi. Taxlashning quyidagi shartlari mavjud:

- varaqlar ustma-ust taxlanadi va hamma varaqlarning markazi bitta nuqtada yotishi kerak;
- taxlangandan keyin varaqlarning har birining kamida bitta nuqtasi (tepadan qaralganda) ko'rinishi kerak;
- u har bir varaqni faqat 90° bura oladi xolos;
- taxlangan barcha varaq tomonlari stol tomonlariga parallel bo'lishi kerak.

U bu masalaga ham tezkorlik bilan dastur ishlab chiqdi. Sizning vazifangiz shu dasturni qaytadan yozish.

Berilganlar:

n natural son ($1 \leq n \leq 100$) varaqlar soni va keyingi n ta qatorning har birda bittadan varaqning o'lchamlari (eni va bo'yi, $1 \leq w, h \leq 10000$).

Natija:

Agar u hamma varaqlarni talab etilganidek taxlay olsa "Yes", aks holda "No" so'zini chiqaring.

<i>Nº</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	4 4 1 10 7 2 5 8 9	Yes
2	5 3 6 5 5 2 4 5 5 1 1	No

8.069

Devor [12].

Doston va uning do'sti Azimboy devor bo'yashga qaror qilishdi. Devor taxtachalardan iborat bo'lib, u 1, 2, 3, 4, ... shu tariqa nomerlangan. Taxtachalarni Doston qizil rang bilan bo'yaydi, Azimboy esa yashil rang bilan. Doston taxtachalarni bo'yashda bitta x sonini oladi va $x, 2*x, 3*x, 4*x...$ shu nomerdagi taxtachalarni bo'yaydi. Azimboy esa y sonini tanlaydi va $y, 2*y, 3*y, 4*y...$ nomerdagi taxtachalarni bo'yaydi. Endi Dostonni bir savol o'ylantirdi $[a, b]$ oraliqdagi taxtachalardan nechitasi bir vaqtda qizil hamda yashil rangga (yani bir taxta ikki xil rangda bo'ladi) bo'yaldi? Siz unga yordam bering.

Berilganlar:

To'rtta butun son x, y, a, b . ($1 \leq x, y \leq 1000, 1 \leq a, b \leq 2 \cdot 10^9, a \leq b$).

Natija:

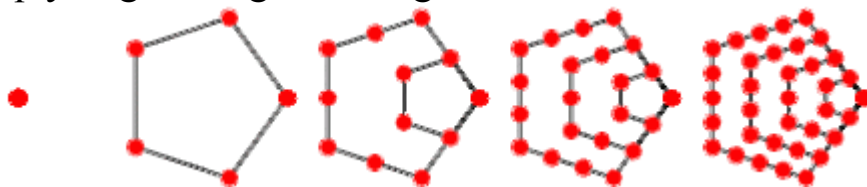
a -taxtadan b -taxtagacha nechta taxta qizilga ham yashilga ham (ikkala rangga ham) bo'yalgan.

Berilganlar:	Natija:
4 6 20 201	15

8.070

Ketma-ketlik [12].

Sizga quyidagicha figura berilgan bo'lsin.



Har bir figuradagi qizil nuqtalar sonini ketma-ketlik ko'rinishida tasvirlasak, quyidagicha ketma-ketlik hosil bo'ladi.

1, 5, 12, 22, 35, ... sizning vazifangiz shu ketma-ketlikning n -hadini topishdan iborat.

Berilganlar:

Natural son ($1 \leq n \leq 10^9$).

Natija:

Ketma-ketlikning n -hadi.

Berilganlar:	Natija:
1	1
3	12
5	35

8.071

XOR [12].

Binar ikkilik amallardan biri bu “*Ikkining moduli bo'yicha qo'shish*” (XOR) amali. C++ va Java dasturlash tillarida XOR amali '^' belgisi orqali amalga oshiriladi. Sizga N natural soni berilgan. Sizning vazifangiz *1 dan N gacha* bo'lgan natural sonlarning XOR amalini, ya'ni quyidagi ifodaning qiymatini hisoblash talab etiladi:

$$1^2^3^{\dots}^N.$$

Berilganlar:

N soni beriladi ($1 \leq N \leq 10^9$).

Natija:

Ifodaning qiymati chiqarilsin.

<i>N</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	1	1
2	10	11

8.072

Yig'indi ko'rinishida ifodalash [12].

Sizga X manfiy bo'lmagan butun son berilgan. Uni shunday n ta manfiy bo'lmagan butun sonlar yig'indisi ko'rinishida yozingki, yig'indida ishtrok etgan eng katta va eng kichik sonlar farqi minimal bo'lsin.

Berilganlar:

Birinchi qatorda X va n butun sonlari beriladi ($0 \leq X \leq 10^9$, $1 \leq n \leq 300$).

Natija:

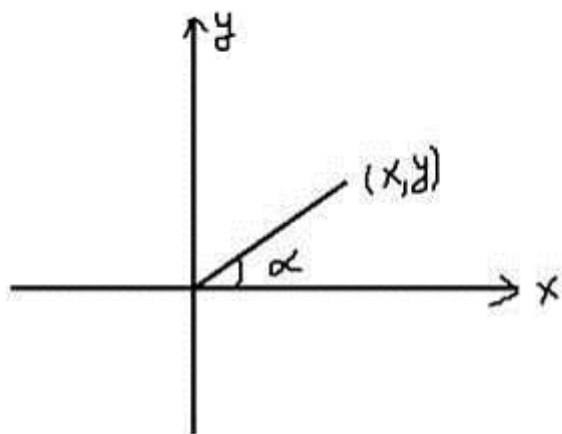
Yig'indida qatnashadigan sonlar qiymati kamaymaslik tartibida orasiga '+' amali qo'yib chiqarilsin.

<i>N</i>	<i>Berilganlar:</i>	<i>Natija:</i>
1	10 3	3+3+4
2	0 4	0+0+0+0
3	5 1	5

8.073

Polyar burchak [12].

Ikki o'lchovli koordinatalar sistemasida (x,y) nuqa berilgan. Ushbu nuqta polyar burchagining qiymatini hisoblovchi dastur tuzilsin.



Berilganlar:

x va y butun sonlari berilgan ($-10^4 \leq x, y \leq 10^4$). Berilgan nuqta koordinata boshida yotmaydi.

Natija:

Qiymati $[0, 360)$ oraliqda bo'lgan polyar burchak, 10^{-4} aniqlikda chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
3 4	53.1301

8.074

Eng ko'p uchragan harf [12].

Sizga s satr berilgan. Sizning vazifangiz unda eng ko'p uchraydigan harfni topish.

Berilganlar:

s sart berilgan. U bo'sh emas va uzunligi 10^5 belgidan oshmaydi. Belgilar kichik lotin alfaviti harflaridan iborat.

Natija:

Eng ko'p uchragan harf chiqarilsin. Agar bunday harflardan bir nechta topilsa, ulardan alfavitda birinchi uchraydigani chiqarilsin.

N_2	<i>Berilganlar:</i>	<i>Natija:</i>
1	Adcda	a
2	Sdzzsws	s

8.075

Eski magnitafon [12].

Magnitafon so'zi grekchadan olingan bo'lib "ovoz" degan ma'noni anglatadi. Magnitafon–elektromexanik qurilma bo'lib, u magnit manbalarga yozilgan signallarni ovoz qilib eshittirishga xizmat qiladi. Magnit manbalarga quyidagilar kiradi: magnit lenta, sim, manjeta, disk va h.k. Hozirda

texnologiyalar rivojlana borgan sari magnitafonlarning turlari ko'payib, ular turlicha nom bilan atala boshladi. Hozirgi masala eski lentali magnitafon bilan bog'liq. Sizga magnitafon o'ragichining dastlabki radiusi $R1$ sm va uning aylanish tezligi V m/s ma'lum. Agar o'ragich T sekundda tasmani o'rab olgan va o'ragichning oxirgi radiusi $R2$ sm bo'lsa, lentaning qalinligini toping (mm). Sizga $R1$, $R2$, T va V haqiqiy sonlari beriladi.

Berilganlar:

$R1$, $R2$, T va V haqiqiy sonlari bittadan bo'sh joy bilan ajrailgan holda berilgan. ($1 \leq R1, R2, T$ va $V \leq 102$)

Natija:

Magnitafon lentasi uzunligi 10^{-5} aniqlikda chiqarilsin.

<i>Berilganlar:</i>	<i>Natija:</i>
1 30 3	0.08378

8.076

Yura va satrlar [12].

Yura satrlar bilan ishlashni yaxshi ko'radi. Shu sababli uning ustozlari Azat Yusupov unga yangi topshiriq berdi. Topshiriq shunday: berilgan S satrdan shunday l va r juftliklarni topish kerakki, bu oraliqda P satrning almashtirishlari (perestanovkasi) mavjud bo'lsin. Masalan: $S = \text{"lshkavatq"}$, $P = \text{"shavkat"}$ bo'lsa, u xolda biz izlayotgan oraliq $[2,8]$ ga teng. Bunday oraliqlar bir necha bo'lsa, barchasini ekranga chiqaruvchi dastur tuzish lozim. Yura bu masalani yechishda qiynalayapti. Unga yordam bering.

Berilganlar:

Birinchi qatorda S ($1 \leq |S| \leq 106$) satr berilgan. Ikkinchi qatorda P ($1 \leq |P| \leq 103$) satr berilgan. S va P satrlar katta va kichik lotin harflaridan iborat.

Natija:

Agar bu oraliqlarni hosil qilish mumkin bo'lsa "YES", aks holda "NO" ni chiqaring. Keyingi qatorlarda hosil bo'lgan oraliqlarni alohida qatorga chiqaring. Agar bu oraliqlar juda ko'p bo'lsa, ulardan dastlabki 200 tasini chiqaring.

<i>Berilganlar:</i>	<i>Natija:</i>
<i>SdotsnoEQ</i>	YES
<i>Doston</i>	2 7

<i>Sdotsneq</i>	<i>NO</i>
<i>Doston</i>	
<i>Aaaaaaaaa</i>	<i>YES</i>
<i>A</i>	<i>1 1</i>
	<i>2 2</i>
	<i>3 3</i>
	<i>4 4</i>
	<i>5 5</i>
	<i>6 6</i>
	<i>7 7</i>
	<i>8 8</i>
	<i>9 9</i>

8.077

O'ylangan son [12].

Kompyuter texnologiyalari rivojlangan asrda sizga yana bir qiziqarli masala. Men 4 xonali son o'ylayman va undan raqamlari yig'indisini ayiraman. Hosil bo'lgan sonning 0 bo'lmagan bitta raqamini o'chirib qolganini aytaman. Sizning vazifangiz o'chirilgan raqamni topish.

Berilganlar:

Uchta butun a , b , c , ya'ni hosil bo'lgan sonning qolgan raqamlari ($0 \leq a, b, c \leq 9$) berilgan.

Natija:

O'chirilgan raqam.

<i>Berilganlar:</i>	<i>Natija:</i>
<i>1 9 0</i>	<i>8</i>
<i>7 8 9</i>	<i>3</i>

Izoh: 1-testda o'ylangan son 1927 raqamlar yig'indisi 19 hosil bo'lgan son 1908. 8 ni o'chiramiz va qolganlari 1 9 0.

8.078

Kubik tashlash [12].

Yon tomonlari 1 dan 6 gacha nomerlangan kubik N marta tashlandi. N marta tashlashda chiqadigan sonlar yig'indisining Q ga teng bo'lish ehtimolligini toping.

Berilganlar:

Ikkita natural son N va Q ($N \leq 500$, $Q \leq 3000$) berilgan

Natija:

Masalaning yechimi 10^{-6} aniqlikda chiqarilsin.

ILOVA: *math.h* kutubxonasiining funksiyalari

Funksiya	Prototipi	Bajaradigan amali
<code>abs(i)</code>	<code>int abs(int i)</code>	<code>i</code> sonining absolyut qiymatini qaytaradi
<code>acos(x)</code>	<code>double acos(double x)</code>	radianda berilgan <code>x</code> argumentning arkkosinus qiymatini qaytaradi
<code>asin(x)</code>	<code>double asin(double x)</code>	radianda berilgan <code>x</code> argumentning arksinus qiymatini qaytaradi
<code>atan(x)</code>	<code>double atan(double x)</code>	radianda berilgan <code>x</code> argumentning arktangens qiymatini qaytaradi
<code>atan2(x)</code>	<code>double atan2(double x, double y)</code>	radianda berilgan <code>x/y</code> argumentning arktangens qiymatini qaytaradi
<code>ceil(x)</code>	<code>double ceil(double x)</code>	haqiqiy <code>x</code> qiymatini unga eng yaqin katta butun songacha aylantiradi va uni haqiqiy ko'rinishini qaytaradi
<code>cos(x)</code>	<code>double cos(double x)</code>	<code>x</code> radianga teng bo'lgan burchakning kosinusini qaytaradi
<code>cosh(x)</code>	<code>double cosh(double x)</code>	<code>x</code> radianga teng bo'lgan burchakning giperbolik kosinusini qaytaradi
<code>exp(x)</code>	<code>double exp(double x)</code>	e^x qiymatni qaytaradi
<code>fabs(x)</code>	<code>double fabs(double x)</code>	haqiqiy sonning absolyut qiymatini qaytaradi
<code>floor(x)</code>	<code>double floor(double x)</code>	haqiqiy <code>x</code> qiymatini eng yaqin kichik songa aylantiradi va uni haqiqiy son ko'rinishida qaytaradi
<code>fmod(x, y)</code>	<code>double fmod(double x, double y)</code>	<code>x</code> sonini <code>y</code> soniga bo'lish natijasidagi qoldiqni qaytaradi. <code>%</code> amaliga o'xshagan, faqat haqiqiy son qaytaradi
<code>frexp(x, expptr)</code>	<code>double frexp(double x, int *expptr)</code>	<code>x</code> sonning mantissasini va darajasining qiymatini ajratib, mantissa qiymatini qaytaradi va darajasini ko'rsatilgan <code>expptr</code> adresiga joylashtiradi
<code>hypot(x, y)</code>	<code>double hypot(double x, double y)</code>	to'g'ri burchakli uchburchakning katetlari bo'yicha gipotenuzasini hisoblaydi

labs(x)	long int labs(long int num)	num uzun butun sonning absolyut qiymatini qaytaradi
ldexp(x)	double ldexp(double x, int exp)	$x \cdot 2^{\text{exp}}$ qiymatni qaytaradi
log(x)	double log(double x)	x sonining natural logarifmini qaytaradi
log10(x)	double log10(double x)	x sonining 10 asosli logarifmini qaytaradi
modf(x)	double modf(double x, double *intptr)	x sonining kasr qismini qaytaradi va butun qismini intptr adresga joylashtiradi
Poly(x,n,c[])	double poly(double x, int n, double c[])	$c[n]x^n + c[n-1]x^{n-1} + \dots + c[1]x + c[0]$ polinomni qiymatini hisoblaydi
pow(x,y)	double pow(double x, double y)	x^y ning qiymatini hisoblaydi
pow10(p)	double pow10(int p)	10^p ni hisoblaydi
sin(x)	double ceil(double x)	x radianga teng bo'lgan burchakning sinusini qaytaradi
sinh(x)	double sinh(double x)	x radianga teng bo'lgan burchakning giperbolik sinusini qaytaradi
sqrt(x)	double sqrt(double x)	x sonining kvadrat ildizini qaytaradi
tan(x)	double tan(double x)	x radianga teng bo'lgan burchakning tangensini qaytaradi
tanh(x)	double tanh(double x)	x radianga teng bo'lgan burchakning giperbolik tangensini qaytaradi

Foydalanilgan adabiyotlar va internet resurslari:

1. Ш.Ф.Мадрахимов, С.М.Гайназаров “С++ тилида программалаш асослари” – Тошкент-2009 й., 196 с.
2. Г.Шилдт –“Полный справочник по С++” – М-2006., 801 стр.
3. О.М.Shukurov, F.Q.Qorayev, E.A.Eshboyev, B.H.Shovaliyev – “Dasturlashdan masalalar to’plami”-Т-2008 у., 160 с.
4. О.М.Shukurov, E.A.Eshboyev, B.H.Shovaliyev – “Delphi va C++ algoritmik tillarida dasturlash” – Qarshi-2012 у., 228 с.
5. М.Э.Абрамян “Электронный задачник по программированию” Ростов - на - Дону 2005 г. 182 стр.
6. В.Н.Пилщиков. “Сборник упражнений по языку Паскаль” М. 1990 г. 120 стр.
7. А.А.Xaldjigitov, Sh.F.Madraximov, U.E.Adambayev, E.A Eshboyev, Informatika va dasturlash. Т.:О‘zМУ, 2005 у, -148 с.
8. Стили и методы программирования. Н.Н.Непейвода Интернет университет информационных технологий. INTUIT.ru, 2005 г., 320 стр.
9. Язык Си и особенности работы с ним Н.И.Костюкова, Н.А.Калинина Интернет-университет информационных технологий. INTUIT.ru, 2006 г., 208 стр.
10. www.acm.timus.ru
11. www.acm.tuit.uz
12. www.algo.urgench-tuit.uz
13. www.codeforces.ru
14. www.cyberforum.ru
15. www.cybern.ru
16. www.delphisources.ru
17. www.cplusplus.com
18. www.acm.dvipion.ru
19. www.acm.sgu.ru
20. www.neerc.ifmo.ru
21. www.acm.msu.ru
22. www.topcoder.com
23. www.olympiads.ru
24. www.ziyonet.uz

MUALLIFLAR HAQIDA



Eshboyev Erkin Abdirashidovich 17-iyul 1974 yilda Qashqadaryo viloyati Chiroqchi tumanining Zarbdor j/x da, ziyoli oilada tug'ilgan. 1997 yildan buyon Qarshi davlat universitetining "Amaliy matematika va informatika" kafedrasida katta o'qituvchi lavozimida ishlab kelmoqda.

E-mail: eea1974@umail.uz



Shodiyev Fayzulla Yusupovich 18-mart 1985 yilda Qashqadaryo viloyati Dehqonobod tumanining Xo'japullast qishlog'ida, ziyoli oilada tug'ilgan. 2011 yildan buyon Qarshi davlat universitetining "Amaliy matematika va informatika" kafedrasida o'qituvchi lavozimida ishlab kelmoqda.

E-mail: fayzullos@mail.ru



Qilicheva Firuza Gulmuradovna 11-avgust 1988 yilda Qashqadaryo viloyati Nishon tumanining Tallimarjon shaharchasida, ziyoli oilada tug'ilgan. 2013 yildan buyon Qarshi davlat universitetining "Amaliy matematika va informatika" kafedrasida o'qituvchi lavozimida ishlab kelmoqda.

E-mail: firuzabonu_11@mail.ru