

Dasturlash bo'yicha olimpiada masalalari va ularni yechishga doir uslubiy ko'rsatmalar. (1-qism)

ID	Party	A	B	C	D	E	F	G	H	I	J	K	L	=	Time	Rank
004	Savaga SAU 2: Муравьев, Антипов, Фидирео	-4	+2	+	-1	+	+3	-3	+	+	+	+	+2	9	1014	1
011	Savaga SAU TEAM 4 and 1/3: Каспарьян	+	+1	+	+1	+	+	+	+	+	+	+3	+	9	1532	2
013	Savaga SU 2: Новиков, Галочкин	+	+	28.24	+	181.11	214.55	281.16	38.34	38.34	+	-7	+3	7	1048	3
001	Savaga SAU JC: Артановов, Назаров, Ульбеков	+	-2	8.25	+	198.44	242.31	258.21	110.11	172.38	+	+1	+	7	1244	4
020	Savaga PSUTI IST1: Смыткин, Исаев, Казаков	+	-2	47.19	+	252.54	280.55	281.25	149.11	89.35	+	+2	+	7	1359	5
029	Savaga LIT BrainSharp: Рыбов, Степанов	-3	-1	31.38	+	138.52	182.31	-7	128.88	125.50	+	+4	+6	7	1465	6
010	Savaga SAU vgs: Волгин, Гостев	+	-12	21.47	-1	140.11	+	+	+1	+1	+	+	+3	6	501	7
007	Savaga SAU GKS: Горбаченко, Кочемаев, Сапожников	+	+	38.24	+	248.21	285.25	+	182.34	140.11	+	+1	+	6	1033	8
012	Savaga SU 1: Мюрсон, Никифоров, Никулин	+	+	38.25	+	140.35	-5	71.31	80.57	+	+	84.41	-1	5	445	9
002	Savaga SAU Rage!: Андреев, Козлов, Саяпин	-3	+	94.01	+	147.81	+	+	+1	+	+	+1	-10	5	477	10
005	Savaga SAU SDK: Скобеев, Михайлов, Бернацкий	+	-3	14.21	+	184.26	-1	+	+2	-3	+	+2	+	4	634	11
014	Savaga SU 3: Моренов, Погилядинов, Казаков	+	-4	87.11	+	-1	+	-2	+2	+	+	-7	+	3	195	12
019	Savaga PSUTI PO1: Федоткин, Подлипнов, Атаев	+	+	17.34	+	84.81	+	+	+1	-2	+	-3	+	3	229	13
008	Savaga SAU 3: Котляров, Круглов, Плетнев	+	+	18.57	+	122.19	+	+	+6	-6	+	+	+	3	391	14
021	Savaga STU 1: Ятульчик, Крайнова, Мухачев	+	+	180.38	+	245.38	-8	+	+1	-1	+	-3	+	3	446	15
009	Savaga SAU HPE: Крестьянников	-5	+	75.41	+	187.38	+	+	+3	-4	+	-7	+	3	552	16
006	Savaga SAU From-nine-to-nine: Матвеев, Просвирнин	+	+	22.46	+	280.38	+6	+	+2	+	+	+	+	3	582	17
023	Savaga STU 3: Слестенин, Осадчий, Головкин	+	+	87.22	-2	188.50	-2	-3	+2	+	+	-3	+	3	583	18
015	Savaga SU 4: Волков, Толстиков, Мальцев	+	+	21.12	+	-2	+	+	+	+	+	-2	+	2	100	19
003	Savaga SAU White spark: Мартынов, Царевский, Молчков	+	+	-6	-2	81.25	282.25	+	+	+	+	+	+	2	263	20
016	Savaga SU 5: Юрьев, Поляков, Кондрашов	+	+	+2	+	+9	+	+	-7	-2	+	+	+	2	569	21
017	Savaga KS_PSUTI 1: Овчинников, Бубнова, Фурса	+	+	28.44	+	+	+	+	-1	+	+	-2	+	1	26	22
018	Savaga PSUTI PO2: Гуртовцев, Пустынникова, Рыков	+	+	82.25	+	+	+	+	-1	-3	+	+	+	1	52	23
022	Savaga STU 2: Волков, Рыбов, Пронин	-3	+	+	+	+	-1	+	+	+	+	-4	+	1	189	24
024	Savaga SUC Непрограммысты: Воловик, Карпейчик, Лисогор	+	+	+3	+	+	-12	+	+	+	+	+	+	1	319	25
025	Savaga SUC IST Kreig: Зайдуллин, Иркина, Халдина	+	+	+	+	+	+	+	+	+	+	-2	+	0	0	26
026	Savaga SUC Локомотив-1: Дычков, Суров, Жаворонкова	-3	+	-2	+	+	+	+	-3	+	+	+	+	0	0	26

O. I. Jalolov, X. U. Xayatov, Z. Z. Shirinov.

**Dasturlash bo`yicha olimpiada
masalalari va ularni yechishga
doir uslubiy ko`rsatmalar.
(1-qism)**

Buxoro 2012 yil

Ushbu uslubiy qo'llanma oliy va o'rta maxsus ta'lim muassasalari hamda dasturlashni o'rganuvchilar uchun mo'ljallangan. Unda internet tizimida olimpiada masalalari va ularni yechish, dasturlashga oid olimpiada saytlari va ulardan acmp.ru sayti, bu saytdagi 5 ta mavzuga doir masalalar va ularning java dasturlash tilidagi dasturlari keltirilgan.

Uslubiy qo'llanmadan talabalar va mustaqil shug'ullanuvchilar internet tizimidagi dasturlash bo'yicha olimpiada masalalarini yechishda kerakli saytlardan foydalanish jarayonlarini, java dasturlash tili yordamida olimpiada masalalarini yechishda tilning imkoniyatlaridan foydalanish usullarini, olimpiada masalalarini yechish algoritmlarini va yechilgan masalalarni olimpiadaga oid saytlarga yuborib to'g'riligini tekshirib ko'rishni o'rganishlari mumkin.

Mualliflar: O.I. Jalolov – Buxoro davlat universiteti “Amaliy matematika va axborot texnologiyalari” kafedrası katta o'qituvchisi.

X.U. Xayatov – Buxoro davlat universiteti “Amaliy matematika va axborot texnologiyalari kafedrası” katta o'qituvchisi.

Z.Shirinov – Buxoro davlat universiteti “Amaliy matematika va informatika” ta'lim yo'nalishi 4-kurs talabasi.

Taqrizchilar: M. Nurullayev – BuxYMTI “axborot texnologiyalari” kafedrası katta o'qituvchisi.

T.B. Boltayev - Buxoro davlat universiteti “Amaliy matematika va axborot texnologiyalari” kafedrası f.m.f.n, dotsenti.

Ushbu uslubiy qo'llanma Buxoro davlat universitetining “Amaliy matematika va axborot texnologiyalari” kafedrasining 2012 – yil 16 – maydagi yig'ilishiga hamda Fizika – matematika fakultetining 20__ - yil __ - ____dagi __ - sonli Kengash yig'ilishida ko'rilgan va nashrga tavsiya qilingan.

I-BOB. Internet tizimida olimpiada masalalari va ularni yechish.

1.1. Dasturlash bo'yicha olimpiadaga oid saytlar.

Hozirgi kunda dasturlashga bo'lgan talab juda katta, hayotdagi har qanday masala dasturlash tili orqali yechiladi. Kompyuterdagi barcha dasturiy taminotlar yoki programmalar albatta dasturlash tillari orqali yaratiladi. Dasturlash bo'yicha qancha ko'p masala bajarsa uning shuncha qobiliyati rivojlanadi. Yechilgan masalani to'g'riligini tekshirish ancha vaqtni oladi. Shuning uchun hozirgi kunda internet tarmog'ida bir qancha olimpiadaga oid saytlar yaratilgan bo'lib, unda masalalar berib qo'yilgan. Foydalanuvchi biror bir masalani yechib, sayt orqali uning to'g'riligini bir necha soniyada tekshirib ko'rishi mumkin. Internet tarmog'ida dasturlash bo'yicha olimpiadaga oid bir nechta saytlar yaratilgan. Bularga quyidagi acmp.ru, neerc.ifmo.ru, acm.timus.ru, www.e-olimp.com, algotist.manual.ru, acm.albertina.ru hamda o'zbek tilidagi acm.tuit.uz saytlarni misol keltirish mumkin. Bu saytlarni yaratishdan maqsad foydalanuvchilar mustaqqil shug'ullanishlari, o'z bilimlarini mustahkamlash uchun imkoniyat yaratish va dasturlashga bo'lgan qiziqishlarini oshirishdan iborat. Ya'ni foydalanuvchi biror bir saytga o'zini registrasiya qilib(ro'yxatdan o'tib) so'ngra shu saytdagi masalalarni mustaqil o'zi yechib uni to'g'riligini tekshirish imkoniyatiga ega bo'ladi. Saytdagi masalalarni dasturlash tillaridan paskal dasturlash tili, Delphi, C++, Visual C++ yoki Java dasturlash tillaridan ixtiyoriy birortasida bajarishlari mumkin. Hozirgi kunda talabalar yoki mustaqil shug'ullanuvchilar uchun o'zbek tilida deyarli adabiyotlar mavjud emas, shularni hisobga olib bitiruv malakaviy ishda o'zbek tilida uslubiy qo'llanma yaratish masalasi qo'yildi. Biror bir dasturlash tilini mukammal o'rgangan va mustaqil o'zi murakkab masalalarni yecha oladigan kishining fikrlash doirasi va tasavvuri keng bo'ladi.

Bitiruv malakaviy ishda biz acmp.ru sayti to'g'risida ma'lumotlar keltiramiz va Java dasturlash tili yordamida masalalarni bajarib uni to'g'riligini sayt orqali yuborib tekshirishni qarab chiqamiz. acmp.ru saytiga kiramiz(1- rasm). Quyida saytning ko'rinishi keltirilgan.



1-рasm. acmp.ru saytini umumiy ko'rinishi.

Hozircha bu saytdan faqat masalalarni ko'rish mumkin. Masalalarni yechib tekshirib ko'rish imkoniyati mavjud emas. Masalalarni yechib tekshirib ko'rish uchun registratsiya bo'limidan ro'yxatdan o'tib login va parolga ega bo'lamiz. Ro'yxatdan o'tish uchun registratsiya bo'limiga kiramiz (2-rasm).

2-рasm. Registratsiya.

O'zingiz to'g'risida barcha ma'lumotlarni kiriting. Ya'ni 2- rasmda keltirilgan bo'sh joylarni to'ldiring va registratsiya bo'limini tanlang. Endi siz login va parolga ega bo'ldingiz xohlagan vaqtda bu saytda login va parolni kiritib masalalarni yechishingiz mumkin (3- rasm).

[новости] [гостевая книга] [форум] [чат]				STUDENT: Shirov Z.Z. Выход						
ИНФОРМАЦИЯ		АРХИВ ЗАДАЧ								
- О школе - Олимпиады - Фотоальбом - Паспорт - Мои задачи - Спонсорам - Архив олимпиад		Поиск: Тема:		Все OK		(Все задачи на одной странице)				
		№1 - №50 из 600					Вперед »			
		C	ID	Задача	Тема	Разбор	Форум	Сложность	Решаемость	Принято
		+	0108	Неглубокий телефон	Задачи для начинающих	Да	0	1%	96%	10241
		+	0001	A+B	Задачи для начинающих	Да	0	2%	81%	17642
		+	0025	Больше-меньше	Задачи для начинающих	Да	0	3%	95%	10665
		+	0195	Эния	Задачи для начинающих	Да	0	3%	96%	7949
		+	0033	Два бандита	Задачи для начинающих	Да	0	4%	92%	9338
		+	0021	Зарплата	Задачи для начинающих	Да	0	4%	92%	9035
		+	0004	Игра	Задачи для начинающих	Да	0	4%	97%	10075
		+	0008	Арифметика	Задачи для начинающих	Да	0	5%	96%	9214
ЗАДАЧНИК		+	0061	Баскетбол	Задачи для начинающих	Да	0	5%	92%	6845
- Архив задач - Состояние системы - Работа в системе - Рейтинг - Разбор задач - Добавить задачу		+	0539	Торт	Задачи для начинающих	Да	0	6%	73%	2480
		+	0092	Журавлики	Задачи для начинающих	Да	0	7%	94%	7308
		+	0106	Монетки	Задачи для начинающих	Да	0	8%	95%	6378
		+	0003	Пятью пять - двадцать пять!	Целочисленная арифметика	Да	0	8%	83%	8646
		+	0149	Разворот	Задачи для начинающих	Да	0	9%	93%	5862
МЕТОДИКА		-	0597	Внезапные гости	Задачи для начинающих	Да	0	10%	86%	2257
- Новичкам - Алгоритмы - Курс К.Шини - Курс олимпиадника - Дистрибутивы - Ссылки		+	0324	Четырехзначный палиндром	Задачи для начинающих	Да	0	10%	95%	5552
		+	0046	Число E	Задачи для начинающих		0	10%	82%	4228
СТАТИСТИКА										
Добавить на Яндекс										

3-rasm. Login va parolni kiritib masalalarni ko'rish.

Bu saytda barcha masalalar arxiv zadach bo'limida joylashtirilgan. Arxiv zadach bo'limini tanlasak 3-rasmda keltirilganidek barcha masalalar chiqadi. Foydalanuvchi bu masalalarni mavzu bo'yicha yoki qiyinlik darajasi bo'yicha saralab olishi mumkin. Buning uchun tema bo'limidan ixtiyoriy mavzu tanlanadi natijada faqat shu mavzuga oid masalar chiqadi. Agar slojnost ustuniga sichqonchani bossak masalalar qiyinlik darajasi bo'yicha saralanadi.

1.2. Java dasturlash tilida olimpiada masalalarini yechish va tekshirish.

Saytdagi barcha masalalarda qiymatlar faylda berilib natija ham faylda chiqariladi. Shuning uchun biz bu qismda Java dasturlash tilidan foydalaniladigan operatorlarni va fayl bilan ishlash imkoniyatlarini batafsil keltiramiz va oxirida biror bir olimpiada masalasini Java dasturlash tilida yechib uni yuborishni ko'ramiz.

Java dasturining strukturasi.

```
class < dastur nomi > {
    public static void main(String args [ ]) {
        < Operatorlar ketma-ketligi >
    }
}
```

Biz hozir Java tilida yozilgan dasturni ko'rsatamiz. Misol uchun **Yangi.java** faylini ko'ramiz.

```
class Yangi {  
    public static void main(String args [ ]) {  
        System.out.println("Java tilidagi dastur");  
    }  
}
```

Java tiplari.

1. Oddiy tiplar. 2. Murakkab tiplar.

1. Oddiy tiplar.
2. Butun tiplar.
3. Haqiqiy tiplar.
4. Simvolli(Harfii) tiplar.
5. Mantiqiy tiplar.

Butun tiplar.

Nomi	Razryadi	Chegarasi
Long	64	-9, 223, 372, 036, 854, 775, 808 9, 223, 372, 036, 854, 775, 807
Int	32	-2, 147, 483, 648 2, 147, 483, 647
Short	16	-32, 768 32, 767
byte	8	-128 127

1. Ifodadagi barcha operandlar eng kami *int* ga keltiriladi. Agar ifodada *long* operand ishtirok etgan bo'lsa, ifodaning tipi *long* bo'ladi.
2. Agar ifodaning tipi majburiy o'zgartirilsa buning uchun ifodaning qiymati u o'tadigan tip diapazoniga tushishi kerak, aks holda ifodaning qiymati o'tish kerak bo'lgan tip moduliga bo'linib qoldiq olinadi.
3. Barcha literallar *int* tipini oladilar. Agar literalning qiymati *long* diapazoniga o'tsa literalning oxirida l(L) harfi yoziladi.

Misol:

int i; long j; short k; byte y; // Har xil butun tipli o'zgaruvchilar e'lon qilindi.

i=-2147483648; j=-9223372036854775808L; k=32767; y=127;

// butun tipli o'zgaruvchilarga qiymatlar o'zlashtirilyapti.

System.out.println("i="+i+"; j="+j+"; k="+k+"; y="+y+");

Dastur bajarilganidan keyingi natija:

i=-2147483648; j=-9223372036854775808; k=32767; y=127; **Haqiqiy tiplar.**

Nomi	Razryadi	Chegarasi
Double	64	1. 7e-308 1. 7e+ 308
Float	32	3. 4e-038 3. 4e+ 038

Harfiy tip(char).

Izoh: *Char* tipi uchun xotiradan 2 bayt ajratiladi va u ixtiyoriy simvolni saqlaydi.

Boshqacha qilib aytganda shu simvolning UNICODE dagi kodini saqlaydi. Ammo bu qiymat bilan ixtiyoriy arifmetik operatsiyatsiyalarni bajarish mumkin. Agar operatsiyaning natijasi yana *char* tipida saqlansa u UNICODE dagi qandaydir kod deb tasavvur kilinadi. Bu qiymat 0 ÷ 65535 bo'lishi shart.

Mantiqiy tip(Boolean).

Mantiqiy tipning 2 xil qiymati mavjud:

1. false(yolg'on).
2. true(chin).

Misol:

boolean t=true; // E'lon kilish:

if(t) System.out.println("Chin"); else System.out.println("Yolg'on");

Dastur bajarilganidan keyingi natija: Chin

Murakkab tiplar.

Satr tipi (String).

Misol:

String S,S1,S2; // E'lon kilish:

S1="Maktab"; S2=S1+"N4"; System.out.println(S2);

Dastur bajarilganidan keyingi natija: MaktabN4

Massiv tipi.

Massiv deb bir necha (soni chegaralangan) bir xil tipga ega bo'lgan elementlarning tartiblashtirilgan joylashuviga aytiladi.

```
Int S[]=new int[10]; // E'lon kilish:
```

```
S[0]=12; S[3]=35; // 1- va 4- elementlariga qiymat o'zlashtirish
```

```
long A[][]=new long[4][3]; // Ikki o'lchovli massivni e'lon qilish.
```

To'plam tiplar.

To'plam bu tariblangan elementlar jamlanmasi. Tartiblash har bir to'plam uchun alohida. Masalan:

ArrayList bu to'plam elementlari takrorlanishi mumkin va element qo'shishda oldin kiritilgan element oldinda turadi.

HashSet bu to'plam elementlari takrorlanmaydi va element qo'shishda variatsialash usulidan foydalaniladi.

LinkedList bu to'plam elementlari takrorlanishi mumkin va qo'shilgan element oxirda yoziladi.

TreeSet bu to'plam elementlari takrorlanmaydi va element o'sib borish tartibida joylashtiriladi.

TreeSet to'plamida qo'shilgan elementlarning joylashishini ko'raylik.

```
TreeSet ts=new TreeSet ();
```

```
ts.add("B");
```

```
ts.add("A");
```

```
ts.add("B");
```

```
ts.add("E");
```

```
ts.add("T");
```

```
ts.add("G");
```

```
System.out.println("TreeSet "+ts);
```

Dastur bajarilganidan keyingi natija: TreeSet [A, B, E, G, T]

Arifmetik operatorlar.

Operator	O'qilishi
+	Qo'shish
-	Ayirish
*	Ko'paytirish
/	Bo'lish
%	Qoldikli bo'lish
++	Inkrement

Munosabat operatorlari.

Operator	O'qilishi
==	Teng
!=	Teng emas
>	Katta
<	Kichik
>=	Katta yoki teng
<=	Kichik yoki teng

Mantiqiy operatorlar.

Operator	O'qilishi
&	Mantiqiy va (AND)
	Mantiqiy yoki (OR)
^	(XOR)
	Tezkor yoki
&&	Tezkor va
!	Mantiqiy inkor (NOT)
!=	Teng emas
==	Teng

A	B	OR	AND	XOR	NOT A
false	False	false	false	false	true
true	False	true	False	true	false
false	True	true	False	true	true
true	True	true	True	false	false

Tarmoqlanish operatorlari.

if (<shart >) < operator >

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

switch-case operatori.

```
Switch(<>){  
  case <1-xol >:<1-operatorlar>  
  case <2-xol >:<2-operatorlar >  
  :  
  case <k-xol >:<k-operatorlar >  
  default: <operatorlar >  
}
```

Misol: Raqamlarni so'z orqali ifodalash.

```
String s; s="1"; int ii=11; pw.print(ii+" Raqam ");  
switch(ii){  
case 0:pw.print("Nol"); break;  
case 1:pw.print("Bir"); break;  
case 2:pw.print("Ikki"); break;  
case 3:pw.print("Uch"); break;  
case 4:pw.print("To'rt"); break;  
case 5:pw.print("Besh"); break;  
case 6:pw.print("Olti"); break;  
case 7:pw.print("Yetti"); break;  
case 8:pw.print("Sakkiz"); break;  
case 9:pw.print("to'qqiz"); break;  
default:pw.print("emas");  
}
```

Ishlash prinsipi: Bu yerda <**ifoda**> *byte*, *short*, *int*, *long* yoki *char* tipiga ega. Ifoda hisoblanib qiymati topiladi va bu qiymat < **1-hol**>,< **2-hol**>... < **k-hol**> qiymatlar bilan taqqoslanadi. Faraz qilayliq < i-hol> qiymatiga teng bo'ldi. U holda < **i-operatorlar**> ketma-ketligidan boshlab ishlaydi. Agar < **ifoda**> ning qiymati biror

bir hol bilan mos kelmasa *default* katoridagi operatorlar ketma-ketligi ishlaydi. Shu bilan *switch-case* operatori o'z ishini tugatadi.

Takrorlash operatorlari.

`while (< shart >) < operator >`

Ishlash prinsipi: Bu yerda **<shart>** mantiqiy ifoda. **<shart>** tekshiriladi agar shart chin natija bersa **<operator>** bajariladi va yana **<shart>** tekshiriladi. Agar **<shart>** yana chin natija bersa **<operator>** yana bajariladi va yana **<shart>** tekshiriladi va hokazo toki **<shart>** dan yolg'on natija chiqquncha **<shart>** dan yolg'on natija chiqdiki **<operator>** bajarilmasdan *while* operatori o'z ishini tugatadi.

Misol: 1 dan 100 gacha sonlar yig'indisi.

```
int l=1,n,yi; n=100; yi=0;
```

```
while(l<=n){  
yi=yi+l; l++; }
```

```
System.out.println("Javob-->" + yi);
```

Dastur bajarilganidan keyingi natija: Javob-->5050

do-while takrorlash operatori.

`do <operator> while <shart>`

Ishlash prinsipi: Operator bajariladi va undan keyin **<shart>** bajariladi. Agar **<shart>** ning natijasi chin bo'lsa yana **<operator>** bajariladi va yana **<shart>** tekshiriladi va hokazo toki **<shart>** dan yolg'on natija chiqquncha. **<Shart>** dan yolg'on natija chiqdiki *do-while* operatori o'z ishini tugatadi deb hisoblanadi.

Izoh: Bu yerda **<operator>** hech bo'lmaganda bir marta ishlaydi.

Misol:

```
int l=101,n,yi; n=100; yi=0;
```

```
do{ yi=yi+l; l++; } while(l<=n);
```

```
System.out.println("Javob-->" + yi);
```

Dastur bajarilganidan keyingi natija: Javob-->101

for ([<initsializatsiya>;[<shart>;[<orttirma>]]) <operator>

ishlash prinsipi: **<initsializatsiya>** qism ishlaydi faqat bir marta undan keyin **<shart>** bajariladi. Agar **<shart>**dan chin natija chiqsa **<operator>** bajariladi va undan keyin **<orttirma>** bajariladi yana **<shart>** tekshiriladi. Agar **<shart>**dan chin natija chiqsa yana **<operator>** bajariladi va undan keyin **<orttirma>** bajariladi yana **<shart>** tekshiriladi. Toki **<shart>**dan yolg'on natija chiqquncha **<shart>**dan yolg'on natija chiqdiki *for* operatori o'z ishini tugatadi deb hisoblanadi. Misol: 1 va n sonlari orasidagi sonlar yig'indisi.

```
int l=101,n,yi; n=100;yi=0;
```

```
for(l=-5;l<=n;l++) yi=yi+l;
```

```
System.out.println("Javob-->" + yi);
```

Dastur bajarilganidan keyingi natija: Javob-->5035

Break operatori.

1) **break;**

2) **break <metka >;**

<metka >siz break ikki maqsadda ishlatiladi.

1) *Switch-case* operatorida tarmoq ishini tugatish maqsadida.

2) Siklda sikl ishini tugatish uchun: agar sikllar ichma-ich joylashgan bo'lsa *break* operatori o'zi joylashgan eng ichkaridagi sikl ishini tugatadi.

Metkali *break* operatorini ishlatish uchun shu *break* operator joylashgan biror bir tashqi blok biror bir nom bilan nomlangan bo'lishi kerak. U holda *break <metka>* operatori bajarilishi natijasida shu *break* operatori joylashgan va **<metka>** nomlangan tashqi blok o'z ishini tugatadi.

```
a1:{
:
{
  if(<shart>) break a1;
: }
: }
```

continue operatori.

Bu operator *while*, *do-while*, *for* operatorlari ichida ishlatiladi.

Misol: 1 dan 100 gacha sonlar yig'indisi.

```
for(;l<=100;l++){ if(l==50) continue; else yi=yi+l; }
```

```
System.out.println("Javob-->" + yi);
```

Dastur bajarilganidan keyingi natija: Javob-->5000

continue operatorining ishlashi natijasida siklning *continue* operatoridan keyingi qismi tashlab yuboriladi, ya'ni *while* va *do-while* sikllarida **<shart>** bajariladi va **<shart>**dan chin natija chiqsa siklning keyingi iteratsiyasi ishlaydi aks holda siklning ishi tugaydi. *for* siklida esa **<orttirma>** ishlaydi va undan keyin **<shart>** bajariladi **<shart>**dan chin natija chiqsa siklning keyingi iteratsiyasi ishlaydi aks holda sikl ishi tugaydi.

Metkali continue operatori.

Continue < metka >;

Ichma-ich joylashgan sikllar holatida agar *continue* operatori ichkaridagi siklning ichida joylashgan bo'lsa va u **<metka>**siz bo'lsa u shu ichkaridagi siklni keyingi iteratsiyasini boshlaydi.

Agar tashqi siklga **<metka>** qo'yilgan bo'lsa ichki siklda joylashgan shu **<metka>**li *continue* operatori tashqi siklning keyingi iteratsiyasini boshlaydi.

continue operatori ishlashi natijasida *for* siklining keyingi iteratsiyasi boshlanadi.

return operatori.

return operatori ishlashi natijasida ishlayotgan dastur(metod) darhol uz ishini tugatadi va boshkarish joriy dasturni(metodni) qaysi programma(metod) chakirgan bulsa ushanga uzatiladi. Agar joriy dastur *main* metodi bo'lsa boshkarish operatsion tizimga uzatiladi.

Misol: 1 dan 100 gacha sonlar yig'indisi.

```
int l=1,n,yi; n=100;yi=0;
```

```
for(;l<=100;l++){ if(l==50){ System.out.println("Javob-->" +yi); return; }  
  
else yi=yi+l; }
```

```
System.out.println("Javob-->" +yi);
```

Dastur bajarilganidan keyingi natija: Javob-->1225

Acmp.ru sayti dasturlashga oid olimpiada masalalar to'plami bilan birga, masalalarni avtomatik tekshirishni o'z ichiga olgan tizim hisoblanadi. Tizimda ishlash uchun ro'yxatdan o'tish va masalalar to'plami (["Архив задач"](#)) bo'limiga o'tish kifoya, sizga ayni vaqtda bu bo'limdan 600 ta turli xil qiyinchilikdagi masalalarni yechih taklif qilinadi. Masalalarning qiyinchilik darajasi 1 dan 100 gacha, bu qiyinchilik darajasi qiymatlariga qarab qatnashchilar o'rinlari belgilanadi, qatnashchilar o'rni ["Рейтинг"](#) bo'limida keltirilgan.

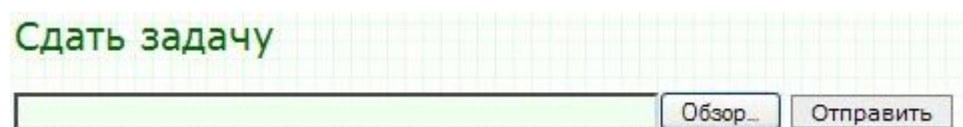
Barcha masalalarni INPUT.TXT va OUTPUT.TXT, berilganlarni o'qish va masalalar javoblarini chiqarishga mo'ljallangan fayllardan foydalanib ishlash talab qilinadi. Masala yechimlarini faqat ro'yxatdan o'tgan foydalanuvchilargina tekshirish uchun yubora oladilar va faqat boshlang'ich kod fayllarining kengaytmalari *.pas, *.dpr, *.c, *.cpp, *.java, *.bas bo'lishi mumkin. Tizim faqat Pascal, C++, Java и Basic, dasturlash tillarida tuzilgan dasturlarni tekshirishga mo'ljallangan. Quyidagi kompilyatorlardan foydalaniladi:

- Borland Delphi 7.0
- Microsoft Visual C++ 7.1
- Java 2 SDK 1.6.0.24
- Microsoft Visual Basic 6.0

Siz yuborgan har bir masala yechimi ma'lum qoidalar asosida 10 dan kam bo'lmagan tahlil uchun maxsus tuzilgan testlar orqali servarda tekshiriladi. Masalalar yechimlari faqatgina tasodifiy olingan testlar orqali hisoblanadi. Mabodo yechim noto'g'ri bo'lsa, testdan o'tkazish to'xtatiladi va xatoni aniqlaydi. Siz ishlagan masalangizning testlar natijalarini ["Состояние системы"](#) bo'limidan

ko'rishingiz mumkin, yana u yerda boshqa foydalanuvchilarning natijalarini ham ko'rishingiz mumkin. Tizim tekshirish natijalarini ma'lum qiladi va natija xato bo'lsa test nomerini ham ma'lum qiladi (agar u biror o'ringa ega bo'lsa).

Yechilgan masalalar barchasi quyidagi bo'lim orqali yuboriladi. Masalaning yechimi joylagan manzillar qatoriga “Обзор” tugmasi orqali yechim manzili ko'rsatiladi. Oxirida “Отправить” tugmasi bosiladi va yechim tizim joylashgan serverga yuboriladi (4-rasm). U yerda yechim maxsus kompyuter orqali avtomatik ravishda tekshiriladi va testlardan o'tkaziladi.



4-rasm. Yechimni yuborish.

№	xabar	Hodisa	Sabab
1	Accepted	Dastur to'g'ri ishladi va barcha cheklovlarga rioya qilish bilan birga kerakli testlardan o'tdi.	Xato va kamchiliklar yo'q
2	Wrong answer	Noto'g'ri javob. Dasturning natijasi bilan hakamlar hayati javobi to'g'ri kelmadi.	Noto'g'ri formatda chiqarilgan yoki dastur algoritmidagi xato bor.
3	Time limit exceeded	Masala uchun belgilangan limit vaqtdan oshib ketilganda. Dastur belgilangan vaqtdan ko'proq vaqt ishladi.	Yechim foydasiz yoki dastur algoritmidagi xato bor.
4	Presentation Error	Hozirda chiquvchi OUTPUT.TXT faylining yo'qligi.	Fayl yaratilmagan yoki boshqa nom bilan yaratilgan.
5	Compilation error	Kompilyatsiya vaqtidagi xatolik. Bu natija kompilyatsiya vaqtida kerakli fayl yaratilmaganligini bildiradi.	Dasturdagi sintaktik xatolik yoki fayl kengaytmasi xato ko'rsatilgan. Balki, dasturlash tili java bo'lganda uning

			boshqatillardan farqli Main sinfining bo'lishi yani konpilyatsiyadan o'tgandan so'ng Main sinfi o'rniga boshqa sinf hosil qilingan bo'lishi ham mumkin.
6	Memory limit exceeded	Masala uchun belgilangan limit xotira hajmidan oshib ketildiganda. Dastur ishlash vaqtida xotiradan ko'p joyni egallashi mumkin.	Foydasiz algoritmdan foydalanilgan.
7	Runtime error	Bajarilish vaqtidagi xatolik. Dastur ishini yakunlashi bilan nolga teng bo'lmagan kod qaytarsa. Bu hodisa dastur natijasini tekshirish imkoni bo'lmasa.	Balki bu xatolik, dastur bajarilganda massivning mavjud bo'lmagan elementi chaqirilganda, nolga bo'lish amali bajarilganda va boshqa hollarda kelib chiqishi mumkin. Dastur C++ tilida bo'lsa va "return 0" operatori bilan tugallanmagan bo'lsa.
8	Compiling	Dasturlar kompilyatsiyadan o'tmoqda.	Ayhi vaqtda fayl bajarilayotgan bo'lsa kerak.
9	Running	Dastur bajarilyapti.	Dastur hozirda qaysi testga kelganligi xabar qilib borladi.
10	Waiting	Kuting.	Dastur testdan o'tayotgan dasturlar arasida navbatini kutmoqda, bir ozdan so'ng tizim uni ham tekshirib ko'ra boshlaydi.

--	--	--	--

II. BOB. Mavzulashtirilgan olimpiada masalalari.

2.1. Boshlovchilar uchun (Начинающих).

Tartib raqami №108

Eshitmaydigan telefon

Siz “Eshitmaydigan telefon” o’yini haqida eshitgan bo’lsangiz kerak. Bu o’yinda ishtirokchilar turli yo’llar orqali bir-birlariga ma’lumot uzatadilar: so’zlar orqali, ko’rgazmali tarzda, hatto yozuvni chap qo’lda yozib boshqa ishtirokchilarga uzatiladi. Yana shunisi ma’lumki, uzatilayotgan ma’lumot deyarli hech qachon oxirgi manzilgacha yetib bormaydi. $F_i(x)$ funksiya sifatida uzatilayotgan x ma’lumot matnini $(i+1)$ -ishtirokchi i -ishtirokchidan qabul qilib oladigan shaklga keltiruvchi funksiyani belgilaylik. U holda oxirgi ishtirokchi ushbu formula bilan ifodalanadigan y ma’lumotni oladi:

$$y = F_{n-1}(F_{n-2}(\dots F_2(F_1(x))))$$

Lekin siz barcha tashqi faktorlarni e’tiborga olmasdan, “eshituvchi telefon” dasturini tuzing. U kiritilgan qiymatlarni hech bir o’zgarishsiz chop qilsin, ya’ni 1 dan $n-1$ gacha bo’lgan i lar uchun $F_i(x)=x$ bo’lsin.

Kiritish: da 1 dan 100 gacha bo’lgan natural son kiritiladi.

Chiqarish: kiritiluvchi fayldagi sonning aynan o’zini chop etsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	5	5

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
import java.util.*;
import java.io.*;
class Main{
public static void main(String asd[])throws IOException
```

```

{
PrintWriter pw;
Scanner sc;
sc=new Scanner(new File("input.txt"));
pw=new  PrintWriter(new File("output.txt"));
int n=sc.nextInt();
pw.print(n);
sc.close();
pw.close();
}}

```

Izoh: Qolgan barcha masalalarning *java* dasturlash tilidagi asosiy yeshim qismini keltiramiz. Ya'ni quyidagi qism barcha dasturlarda bir xilda bo'ladi:

```

import java.util.*;
import java.io.*;
class Main{
public static void main(String asd[]) throws IOExceptio
n{
PrintWriter pw;
Scanner sc;
sc=new Scanner(new File("input.txt"));
pw=new          PrintWriter(new File("output.txt"));
...
...
...
sc.close();
pw.close();
}}

```

Demak `import java.util.*;` va `import java.io.*;` qatorlarida *util* va *io* paketlarining barcha sinflari chaqiriladi. `class Main` bu yerda

Main sinf nomi va *acmp.ru* saytining asosiy sharti sinfning nomi Main bo'lishi talab etiladi.

```
public static void main(String asd[])throws IOException
Scanner(new File("input.txt")); bu yerda "input.txt" fayli
o'qish uchun ochilyapti, (new File("output.txt")); "output.txt"
fayli esa yozish maqsadida ochilyapti. sc.close(); pw.close(); sc va
pw yopilyapti. throws IOException mavjud xatoliklar haqida xabar beradi.
main metodi joylashgan sinf asosiy sinf hisoblanadi, public main metodini
islatishga ruxsat beradi, static main metodining statik ekanligini, void esa
metodning hech qanday natija qaytarmasligini bildiradi. String asd[]
kiritiladigan ma'lumotlarni satr tipli massivga joylashtiradi va ulardan foydalanish
imkoniyati mavjud bo'ladi. Dasrurlarda foydalanilgan // belgi, belgidan keyingi
shu qatorda joylashgan barcha yozilganlar izoh ekanligini va ularni kompiatr
o'qimasligini bildiradi. /* belgidan */ belgigacha yozilganlar ham izoh hisoblanadi.
```

Tartib raqami №264

Issiq havo

Har bir kunda harorat 0°C dan yuqori bo'lgan davrlar ichida eng uzun shunday kunlar zanjiridagi kunlar sonini chiqaring.

Kiritish: 1-satrdan N ($1 \leq N \leq 100$), keyingi satrda bo'sh joy orqali N ta butun son yozilgan bo'lib, har bir son mos kunning haroratidir. Harorat – butun son bo'lib, -50dan 50 gacha bo'lgan oraliqda joylashadi.

Chiqarish: Eng uzun harorati 0°C dan yuqori kunlar zanjirining uzunligi chiqarilsin. Agar har kuni harorat nomusbat (0 dan katta emas) bo'lgan bo'lsa, natija sifatida 0 ni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	6 -20 30 -40 50 10 -10	2
2	8 10 20 30 1 -10 1 2 3	4
3	5	0

	-10 0 -10 0 -10	
--	-----------------	--

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int n=sc.nextInt(),h=0,x=0;
while(sc.hasNext()){ n=sc.nextInt();
if(n>0) x++; else { if(x>h) h=x; x=0; } }
if(x>h) h=x; pw.print(h);
```

Tartib raqami №496

Qoraqat yig'ish

Kareliyada fermer xo'jaligida qoraqat o'stiriladi. U dumaloq qatorlarda o'sadi, bunda tuplar faqat aylana bo'ylab ekilgan. Har bir tupning faqat 2 ta qo'shnisi bor. Qatorda hammasi bo'lib N ta tup mavjud. Bu tuplarning hammasi har xil hosildorlikka ega shuning uchun i - tupda a_i miqdorda meva pishadi.

Bu fermer xo'jaligida qoraqatni avtomatik yig'ib olish tizimi o'rnatilgan. Bu tizim boshqaruvchi modul va bir nechta hosil yig'uvchi modullardan iborat. Hosil yig'uvchi modul biror tupning yoniga kelgach, bir urinishda shu tup va uning qo'shnilarining hosillarini yig'ib oladi.

Kiruvchi faylda qator berilganda hosil yig'uvchi modul bir urinishda yig'ishi mumkin bo'lgan mevalarning eng katta sonini aniqlang.

Kiritish: 1-satrida N ($3 \leq N \leq 1000$) – qoraqat tuplari soni berilgan. 2-satrida esa N ta butun sonlar $a_1, a_2, \dots, a_N$ ni saqlaydi – mos tupda o'sadigan qoraqat mevalari soni. Hamma a_i lar 1000 dan katta emas.

Chiqarish: Dastur natijasi chiqarilsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	4 1 2 3 4	9
2	3 1 2 3	6

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int n=sc.nextInt(),h=0,x=0,i,j;
int a[]=new int[n+2];
```

```

for(i=0;i<n;i++)a[i]=sc.nextInt();
a[n]=a[0]; a[n+1]=a[1]; j=0;
// Boshlang'ich ikki tup oxirgi tupning ikki qo'shnisi bo'la oladi.
while(j<n){
    x=a[j]+a[j+1]+a[j+2]; // x da qo'shni tuplar hosili yigilyapti.
    if(x>h)h=x; x=0; j++; }
    // x va h taqqoslanayti h eng ko'p hosil qiymati.
pw.print(h);

```

Tartib raqami №387

Chap rekursiya

Formal grammatika va avtomatlar nazariyasi(FGAN)da *kontekstli-erkin grammatikalar* (KE-grammatikalar) muhim rol o'ynaydi. KE-grammatika deb quyidagi to'rtlikka aytiladi: N noterminal belgilar to'plami, T terminal belgilar to'plami, P qoidalar (produksiylar) to'plami va N to'plamga tegishli bo'lgan S boshlang'ich belgi.

P dagi har bir p produksiya $A \rightarrow a$ ko'rinishga ega, bu yerda A noterminal belgi (A N ga tegishli), a esa terminal va noterminal belgilardan iborat bo'lgan satrdir. So'zlarni chiqarish jarayoni faqat S boshlang'ich belgi bilan boshlanuvchi satrlardan boshlanadi. Shundan so'ng har qadamda joriy satrga kiruvchi noterminal belgilardan biri o'zi chap qismi sifatida tarkibiga kiradigan produksiyalardan birortasining o'ng qismi bilan almashtiriladi. Agar bu amaldan so'ng faqat terminal belgilardan iborat satr hosil bo'lsa, chiqarish jarayoni tugallanadi.

Ko'pchilik nazariy masalalarda grammatikalarning *normal shakli* ni ko'rib chiqish qulayroq. Grammatikani normal shaklga keltirish chap rekursiyani yo'qotishgan boshlanadi. Ushbu masalada biz bevosita chap rekursiya deb ataluvchi xususiy holni ko'rib chiqamiz. Aytiladiki, agar $A \rightarrow R$ chiqarish qoidasi chap rekursiyaga ega deyiladi, agar R satrning 1-belgisi A bo'lsa.

KE grammatika berilgan. Bevosita chap rekursiyaga ega bo'lgan qoidalar sonini aniqlash talab etiladi.

Kiritish: 1-satrida n ($1 \leq n \leq 1000$)- grammatikadagi qoidalar soni keltirilgan. Keyingi n ta satrlarning har birida 1 tadan qoida kiritilgan. Noterminal belgilar lotin alifbosining bosh harflari bilan, terminallari esa kichik harflari bilan belgilanadi. Produksiya (qoida)ning chap qismi o'ngidan -> belgisi bilan ajratiladi. Produksiyaning o'ng tarafi uzunligi 1 dan 30 tagacha belgidan iborat bo'lishi mumkin.

Chiqarish: Masalaning javobi chiqarilsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	3 S->Sabc S->A A->AA	2

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int  n,min,max,a,b,i=0,c; String s;
min=sc.nextInt();
while(sc.hasNext()){ s=sc.next();
if(s.charAt(0)==s.charAt(3)) i++;
} pw.print(i);
/* Shartda keltirilganlardan shu narsa ma'lum bo'ladiki beriladigan satrlarning 1-
elementi bilan 4- elementlari teng bo'lsa unda berilgan qoida chap rekursiyani
tashkil qiladi. */
```

Tartib raqami №293

Soliqlar

Qaysidir davlatda N ta o'zaro raqobatchi firmalar faoliyat yuritadi. Har bir firma o'z daromadiga ega bo'lib, uning daromadi $V[i]$ so'mga teng. Har bir firmadan alohida soliq olinadi, har bir i -firmaga soliq $p[i]$ foiz bilan belgilanadi. Qaysi firma eng ko'p soliq to'lashini hisoblang.

Kiritish: Dastlab N ($0 < N \leq 100$) – firmalar soni, so'ngra har biri 154 dan oshmaydigan N ta butun nomanfiy sonlar – firmalarning daromadi, keyin esa yana N ta 0 dan 100 gacha bo'lgan butun sonlar – firmlarning soliqlari foiz miqdori berilgan.

Chiqarish: 1 ta son- eng ko'p soliq to'lovchi firma nomerini chiqaring. Agar bunday firmalar bir nechta bo'lsa, eng kichik nomerli firmani chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	1 1 1	1
2	2 1 2 3 2	2
3	3 100 1 50 0 100 3	3

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String v=sc.nextLine();
int n=Integer.parseInt(v);
int a[]=new int[n];
int i=0,k=0;
int b[]=new int[n];
String c=sc.nextLine();
Scanner sc1=new Scanner(c);
while(sc1.hasNext()){
a[i]=sc1.nextInt(); i++;}
c=sc.nextLine();
Scanner sc2=new Scanner(c); i=0;
while(sc2.hasNext()){
b[i]=sc2.nextInt();
i++; }
double x=-1;
for(i=0;i<n;i++)
if(b[i]*a[i]*1.0/100>x)
{x=1.0*b[i]*a[i]/100; k=i; }
pw.print(k+1);
```

Tartib raqami №68

Uy-maktab-uy

Vali har kun metroda yuradi. Ertalab maktabga boradi, kechqurun uyga qaytadi. Pulni tejash uchun X ta metroda yurishga mo'ljallangan elektron smart-karta sotib oladi. Bu kartaning metroda yurishlari soni 0 ta bo'lmasa u bimalol turniketdan o'ta oladi va kartadan 1 ta yurish ruxsatnomasini o'chiradi. Agar kartada ruxsat qolmagan bo'lsa, turniket Valini o'tkazmaydi va u yangi karta sotib olishga va yana turniketdan o'tishga majbur bo'ladi.

Vali e'tibor berdiki, ertalab metroda odam ko'pligi sababli u yangi karta sotib olmoqchi bo'lsa maktabga kechikadi. Shuning uchun u aniqlamoqchi, ertalab kartasida yurishga ruxsat qolmaydigan kun bormi yoki yo'qmi?

Vali faqat maktabga boradi va metroga faqat uyi hamda maktabi yonidagi stansiyalardan chiqadi.

Kiritish: 2 ta satr joylashadi. 1-satrdagi Vali kartani qayerdan sotib olganiga qarab "Maktab" ("School") yoki "Uy" ("Home") so'zlaridan biri beriladi, 2-satrdagi esa X natural soni, $1 \leq X \leq 1000$ yoziladi.

Chiqarish: Agar ertalab Valining kartasida 0 ta ruxsat qoladigan kun mavjud bo'lsa, "Ha", aks holda "Yo'q" javobi chop etilsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	Home 1	Yes
2	School 2	No

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s=sc.next();
int n=sc.nextInt();
if(s.equals("School")) { if(n%2==1) pw.print("Yes");
else pw.print("No"); }
else pw.print("Yes");
```

TURIST

G'ani maktabning turistik yig'ilishiga tayyorlanmoqda. U o'z sinfida palatkalar uchun mas'uldir. Uyidan 3 ta palatka topdi: 1-si a_1 kg og'irlikka ega va b_1 ta odamni sig'iradi, 2-si a_2 og'irlikka ega va b_2 ta odamni sig'iradi, 3-si a_3 kg og'irlikka ega va b_3 ta odamni sig'iradi.

G'anining sinfida K ta o'quvchi bor. U palatlarni hammani sig'iradigan qilib qurishi mumkinmi? Bunda tanlangan palatkalar og'irligi birgalikda W kg dan oshmasligi kerak.

Kiritish: 1-satrdan 2 ta K va W butun sonlar berilgan ($1 \leq K \leq 15$, $1 \leq W \leq 30$). 2-satrdan 6 ta butun sonlar kiritilgan: $a_1, b_1, a_2, b_2, a_3, b_3$ ($1 \leq a_1, a_2, a_3 \leq 10$, $1 \leq b_1, b_2, b_3 \leq 15$).

Chiqarish: Agar palatlarni kerakli qurish mumkin bo'lsa, HA("YES") javobini, aks holda YO'Q("NO") javobini chop eting.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	10 10 5 5 6 6 4 5	YES
2	10 10 5 5 6 6 7 7	NO

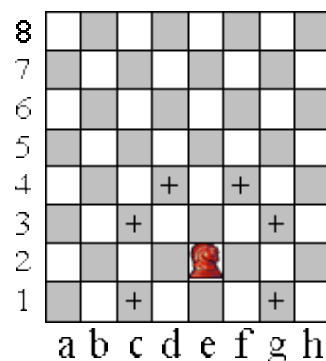
Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int a[]=new int[3]; int b[]=new int[3]; int w[]=new int[5]; int k[]=new int[5];
int n=sc.nextInt(),n1=sc.nextInt();
int i,x=0,x1=0; for(i=0;i<3;i++){ a[i]=sc.nextInt(); x=x+a[i]; b[i]=sc.nextInt();
x1=x1+b[i]; w[i]=x; k[i]=x1; }
w[3]=a[0]+a[2]; w[4]=a[1]+a[2]; k[3]=b[0]+b[2]; k[4]=b[1]+b[2]; int f=0;
for(i=0;i<3;i++)if((n1>=a[i])&&(n<=b[i])) {f++; break;}
for(i=0;i<5;i++) if((n1>=w[i])&&(n<=k[i])) {f++; break;}
if(f!=0) pw.println("YES"); else pw.println("NO");
```

Otning yurishi

Vali shaxmatni o'rganmoqchi, biroq ozgina qiynaladi. Unga shaxmatdagi otning qaysi kataklarga yurishi mumkinligini o'rgatuvchi dastur tuzib bering.

Bilamizki, shaxmatda ot gorizontal 2ta katak va vertikal 1 ta katakka, yoki bo'lmasa 1 ta gorizontal va 2 ta vertikal katakka yura oladi. Vertikallar lotin alifbosining a – h kichik harflari bilan, gorizontallar esa 1 dan 8 gacha bo'lgan raqamlar bilan belgilanadi. Shaxmat doskasidagi ixtiyoriy katak mos vertikal harf va gorizontal raqam orqali belgilanadi.



Kiritish: 2 ta belgi – ot turgan katak koordinatalari beriladi.

Chiqarish: Ixtiyoriy tartibda ot yurishi mumkin bolgan barcha kataklar koordinatalari chiqarilsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	e2	c1 c3 d4 f4 g1 g3
2	a1	b3 c2

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s=sc.next(),c;
c="abcdefgh"; int i=0,t,k=0;
for(i=0;i<c.length();i++)
if(s.charAt(0)==c.charAt(i)){ k=i; break;}
t=(s.charAt(1)-'0');
if((k-2>=0)&(t-1>0))
```

```

pw.println(c.charAt(k-2)+" "+(t-1));
if((k-2>=0)&(t+1<9))
pw.println(c.charAt(k-2)+" "+(t+1));
if((k-1>=0)&(t-2>0))
pw.println(c.charAt(k-1)+" "+(t-2));
if((k-1>=0)&(t+2<9))
pw.println(c.charAt(k-1)+" "+(t+2));
if((k+1<8)&(t+2<9))pw.println(c.charAt(k+1)+" "+(t+2));
if((k+2<8)&(t-1>0))pw.println(c.charAt(k+2)+" "+(t-1));
if((k+2<8)&(t+1<9))pw.println(c.charAt(k+2)+" "+(t+1));
if((k+1<8)&(t-2>0)) pw.println(c.charAt(k+1)+" "+(t-2));

```

Tartib raqami №331

Kelish vaqti

Poyezdning jo'nash vaqti va oxirgi stansiyaga yetgunicha ketadigan vaqt berilgan. Shu poyezdning manzilga yetish vaqtini toping (balki, boshqa sutkaga o'tishi ham mumkin).

Kiritish: 2 ta satr beriladi. 1-sida jo'nash vaqti, 2-sida yo'lga ketadigan vaqt berilgan. Jo'nash vaqti "HH:MM" formatida beriladi, bu yerda HH-soatni bildiradi va 00 dan 23 gacha qiymatga ega bo'ladi; MM-daqiqa, 00 dan 59 gacha qiymat oladi. Yo'lga ketadigan vaqt 2 ta nomanfiy butun sonlar bilan- soatlar va daqiqalar miqdori bilan –beriladi. Sonlar bo'sh joy bilan ajratiladi. Soatlar soni 120 dan, daqiqalar soni 59 dan oshmaydi.

Chiqarish: 1 ta satr bo'lishi kerak – poyezdning oxirgi stansiyaga yetib kelish vaqti chiqarilishi kerak. Bu vaqt formati kiritiluvchi vaqt formati bilan bir xil bo'lishi shart.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	00:00 10 10	10:10
2	01:02	05:08

	4 6	
3	11:00 22 0	09:00

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s=sc.next(),a1,b1;
a1="" +s.substring(0,2); b1="" +s.substring(3);
int a,b,h,f;
a=Integer.parseInt(a1); b=Integer.parseInt(b1);
int x,y,y1,x1; x=sc.nextInt(); y=sc.nextInt(); s="";
h=(y+b)/60; y1=(y+b)-h*60; f=(x+a+h)/24;
x1=(x+a+h)-f*24; if(x1<10) s=s+"0"+x1;else s=s+x1;
if(y1<10) s=s+":0"+y1;else s=s+": "+y1;
pw.print(s);
```

Tartib raqami №409

Temiryo'l

Yangi temir yo'lni qurishda muammolar chiqib qoldi. Yo'l tepalikli joydan o'tadi, biroq temir yo'l qat'iy gorizontal bo'lishi lozim. Shu sababli rahbariyat shu joyda yer yuzasini tekislashga qaror qildi. Asosiy muammo shundaki, 1 kubmetr tuproqni qurilishga olib kelish yoki olib ketish 10000\$ turar ekan. Biroq moddiy bunga qurblari yetmas ekan.

Shu sababli bosh injener shu tepaliklarning o'zidagi tuproqdan foydalanib, yer yuzasini tekislash haqida qaror qaul qildi. Endigi eng murakkab muammo yo'l o'tishi lozim bo'lgan joylarning dengiz sathidan balandligini topishdir.

Kiritish: da 1-satrdan N ($1 < N \leq 30000$) ta balandligi o'lchangan nuqtalar soni berilgan. 2-satrdan esa o'lchashlar natijasi keltirilgan: satrning i -soni uchastka boshidan $(i-1)$ metr masofada joylashgan nuqtaning dengiz sathidan balandligidir. Barcha balandliklar - 10000 dan oshmaydigan nomanfiy butun sonlardir. Qo'shni nuqtalar orasidagi yo'l to'g'ri chiziqli deb oling.

Chiqarish: Javobni 10^{-5} dan kichik bo'lmagan aniqlikda chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	4 0 1 1 0	0.6666666667
2	5 2 2 2 2 2	2.0000000000

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int n=sc.nextInt(),m,s,x,j=0;
double z=0; String d=""; x=sc.nextInt(); z=(double)x/2;
while(sc.hasNext()){ x=sc.nextInt(); z=z+x; }
z=z-x; z=z+1.0*x/2; z=z/(n-1); d="" +z; m=d.length();
int i; if(m>=12){if((d.charAt(12)-'0')<5)
d=d.substring(0,12);
else d=d.substring(0,11)+((d.charAt(12)-'0')+1); }
else for(i=1;i<=12-m;i++)d=d+"0"; pw.print(d);
```

Tartib raqami №315

Eng kichik sanoq sistemasi

Ma'lumki, sanoq sistemasining asosi deb biror sonni shu sanoq sistemasida yozish uchun qo'llash mumkin bo'lgan turli belgilar soniga aytiladi. Shuningdek, ixtiyoriy x soni b -sanoq sistemasida $x=a_0 \cdot b^0 + a_1 \cdot b^1 + \dots + a_n \cdot b^n$ kabi yoziladi, bu yerda $b \geq 2$ va $0 \leq a_i < b$.

$b \leq 36$ bo'lgan sanoq sistemasida sonlarni yozish uchun ushbu 0,1,..., 9, A, B, ..., Z ro'yxatning dastlabli b ta belgisi ishlatiladi. Masalan, sonlarni 3lik sanoq sistemasida yozish uchun 0, 1, 2 belgilari, 12 likda yozish uchun esa- 0,1,..., 9, A, B kerak bo'ladi.

Shunday dastur tuzingki, kiruvchi S satrga qarab bu satr asosi 36 dan oshmaydigan biror sanoq sistemasidagi sonning yozuvimi yoki yo'qligini aniqlash hamda agar shunday bo'lsa, bunday sanoq sistemalaridan eng kichik asosini topsin.

Kiritish: 1 ta satrda kiritiluvchi satr berilgan. Satr uzunligi 255 ta belgidan oshmaydi. Satrning barcha belgilari 32 dan 127 gacha bo'lgan kodga ega.

Chiqarish: 1 ta son chiqarilishi kerak. Agar satr sonning biror sanoq sistemasidagi ko'rinishi bo'lsa, eng kichik asosni chop eting. Aks holda "-1" chiqarilsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	123	4
2	ABCDEF	16
3	AD%AF	-1

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s=sc.nextLine(),c="";
c=c+"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ" ;
int n=s.length(); int i,r,k=0;
for(i=0;i<n;i++){r= c.indexOf(s.charAt(i));
if(r==-1){ k=r; break;} else { if(k<r) k=r;} }
if(k==-1) pw.println(k);
else if(k<=1) pw.println("2"); else pw.println(k+1);
```

Tartib raqami №277

Maktab algebrasi

Ikkita x va y o'zgaruvchili $a+bx+cy$ uchhad a , b va c koeffitsientlari orqali bir qiymatli aniqlanadi. Berilgan a , b va c orqali quyidagi algebraik qoidalarga ko'ra yozilgan uchhadning ko'rinishini chop qiluvchi dastur tuzing:

- agar x yoki y qatnashgan haddagi koeffitsient moduli 1 ga teng bo'lsa, bu koeffitsient tashlab yuboriladi.
- Koeffitsienti 0 ga teng bo'lgan had tashlab yuboriladi (agar barcha koeffitsientlar 0 ga teng bo'lsa, bu uchhad 0 ko'rinishda bo'ladi).
- "+" ishorasi manfiy koeffitsient oldidan kelsa yozilmaydi.
- "+" ishorasi ifodaning boshida kelsa yozilmaydi
- Koeffitsient va o'zgaruvchi orasidagi kopaytirish belgisi yozilmaydi.

Bunda hadlarning o'rnini almashtirish ta'qiqlanadi.

Kiritish: 3 ta koeffitsient lar a, b va c bo'sh joy tashlab yozilgan, ularning har biri absolyut kattaligiga ko'ra 30000 dan katta emas.

Chiqarish: Algebraik qoidalarga asosan yozilgan uchhad chiqariladi.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 2 -1	2x-y
2	3 0 -2	3-2y

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s="";
int a=sc.nextInt(),b=sc.nextInt(),c=sc.nextInt();
if(a!=0)s=s+a;
if(b<-1)s=s+b+"x";else if(b==-1)s=s+"-x";
if((s!="") && (b==1))s=s+"x"; else if(b==1) s=s+"x";
if((s!="") && (b>1))s=s+" "+b+"x"; else if(b>1)s=s+b+"x";
if(c<-1)s=s+c+"y";else if(c==-1)s=s+"-y";
if((s!="") && (c==1))s=s+"y"; else if(c==1) s=s+"y";
if((s!="") && (c>1))s=s+" "+c+"y"; else if(c>1)s=s+c+"y";
if(s=="")s="0";
pw.print(s);
```

2.2. Butun sonlar arifmetikasi (Целочисленное).

Tartib raqami №148

EKUB

Ikkita natural A va B sonlarning eng katta umumiy bo'luvchisini (EKUB) toping.

Kiritish: A va B natural sonlar beriladigan.

Chiqarish: Sonlarning eng katta umumiy bo'luvchisini (EKUB)ni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	12 42	6

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

int a=sc.nextInt(),b=sc.nextInt();
while(a!=b){
//Shart a va b lar teng bo'lgancha bajariladi.
if(a>b) a=a-b; else b=b-a; }
/* sonlar ichida kattasi topilib undan kichigini ayirish orqali EKUB topilyapti. */
pw.print(a);

```

Tartib raqami №327

Baxtdan bir qadam masofada

Bilet nomeri olti xonali sondan iborat birinchi uchta va oxirgi uchta raqamlari yig'indisi bittaga farq qiladi. Shu biletdan bitta oldingi yoki bitta keyingi biletlardan birortasi baxtlimi?(ya'ni birinchi uchta va oxirgi uchta raqamlari yig'indisi tengmi?)

Kiritish: Birinchi qatorda K-testlar soni. Keyingi K ta qatorda biletlar nomerlari. Testlar soni o'ndan ortiq emas.

Chiqarish: Masala yechimiga mos baxtli bo'lsa Yes aks holda No.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	3	Yes
	715068	No
	445219	Yes
	012200	

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

int x,a,q1,q,b,b1,i,n=sc.nextInt();
for(i=0;i<n;i++){
a=sc.nextInt();
x=a; q=0; q1=0; x=x+1;//x, a dan keyingi son.
b=x/1000; // b, x sonning birinchi uchta raqamli qismi.
while(b!=0){
q=q+b%10; b=b/10 ;} //q, b sonning raqamlari yig'indisi.

```

```

b1=x%1000;//b1, x sonning oxirgi uchta raqamli qismi.
while(b1!=0){q1=q1+b1%10; b1=b1/10;}
//q1, b1 sonning raqamlari yig'indisi.
if(q1==q) pw.println("Yes");
else { q=0; q1=0; a=a-1; b=a/1000;
//a ga, a dan oldingi son o'zlashtirilyapti.
while(b!=0){ q=q+b%10; b=b/10 ; } b1=a%1000;
while(b1!=0){ q1=q1+b1%10; b1=b1/10; }
if(q1==q) pw.println("Yes"); else pw.println("No");} }

```

Tartib raqami №22

Birlar

Berilgan sonning ikkilik sanoq sistemasidagi ko'rinishida nechta 1 qatnashishini toping.

Kiritish: N soni kiritilgan n ($0 \leq n \leq 2 \cdot 10^9$).

Chiqarish: Birlar sonini chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	5	2
2	7	3

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

int i=0,l=0,j=0,k=0,w=0,n; long x;
x=Long.parseLong(sc.next());
String s=Long.toString(x,2);
/* java dasturlash tilida Long.toString(x,k); x o'nlikdagi soni k- sanoq sistemasiga
o'tkazadi, natijasi albatta String tipiga tegishli bo'ladi. Hozirda java da shu yo'l
bilan 36 lik sanoq sistemasigacha o'tish mumkin. */
n=s.length();
for(i=0;i<n;i++) w=w+Integer.parseInt((s.charAt(i)+""));
// s satrdagi har bir raqam w soniga ga qo'shilyapti.
pw.print(w);

```

Tartib raqami №147

Fibonachchi soni

Fibonachchi ketma-ketligining n - sonini toping. Quyidagi ketma-ketlik $a_0 = 0, a_1 = 1, a_k = a_{k-1} + a_{k-2} (k > 1)$ Fibonachchi ketma-ketligi deb ataladi.

Kiritish: N natural son kiritilgan ($0 \leq N \leq 30$).

Chiqarish: N - Fibonachchi sonini chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	7	13

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int n=sc.nextInt(), i;
long a[]=new long[n+1]; //a, massiv e'lon qilinyapti.
if(n==0) pw.print(0); else { a[0]=0; a[1]=1;
for(i=2;i<=n;i++) a[i]=a[i-1]+a[i-2];
// a[i] lar Fibonachchi sonlari.
pw.print(a[n]); }
```

Tartib raqami № 6

Shaxmat

Shaxmat doskasida otning yurishi berilgan. Yurishning to'g'ri yoki xato berilganligini aniqlovchi dastur tuzish kerak. Masalan: «C7-D5» yozilsa, dastur to'g'ri yurish deb, «E2-E4» yozilsa noto'g'ri yurish deb aniqlashi lozim. Yurishlar korrekt ifodalanishini ham tekshirish kerak, «D9-N5» yozilsa, dastur “xato yozuv” deyishi lozim.

Kiritish: Otning turgan o'rni va yurish xonasi berilishi mumkin. Belgilar 5 ta dan oshmaydi.

Chiqarish: Yurish to'g'ri bo'lsa, “Yes”, korrekt yozilgan bo'lsayu, noto'g'ri yurish hisoblansa, “No”, agar yozuv noto'g'ri bo'lsa, kooordinatalar berilmasa “Error” degan yozuvni chiqarsin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
---	-----------	------------

1	C7-D5	YES
2	E2-E4	NO
3	BSN	ERROR

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s=sc.next(),s1="",s2="",s3="",ss="",s4="";
int i,a=0,a1=0,a2=0,a3,f3=0,f1=-1,f2=-1,w=0,j=0, f;
s4="ABCDEFGH"; f=s.indexOf('-');
if(f==2){ f1=s4.indexOf(s.charAt(0));
f2=s4.indexOf(s.charAt(3));
if((f1!=-1)&&(f2!=-1)&&
(Integer.parseInt(""+s.charAt(1))>0)&&(Integer.parseInt
(""+s.charAt(1))<9)&&(Integer.parseInt(""+s.charAt(4))>
0)&&(Integer.parseInt(""+s.charAt(4))<9)){
ss=""s.charAt(1); a1=Integer.parseInt(ss);
ss=""s.charAt(4); a2=Integer.parseInt(ss);
if((Math.abs(a1-a2)==2)&&(Math.abs(f1-f2)==1))w++;
else {if((Math.abs(a1-a2)==1)&&(Math.abs(f1-f2)==2))
w++; }
if(w>=1) pw.print("YES");
else pw.print("NO");} else j++; } else j++;
if(j>0)pw.print ("ERROR" );
```

2.3. Geometriya (Геометрия).

Tartib raqami №26

Ikki aylana

Tekislikda ikkita aylana berilgan. Ularning kesishishi yoki kesishmasligini aniqlovchi dastur tuzing. Aylanalarning urinishi ham kesishish deb olinadi.

Kiritish: Ikkita qatorda har birida 3 ta son aylana markazi koordinatalari x va y (butun sonlar va moduli 5000 dan katta bo'lmagan) hamda r radius (natura son $1 \leq r \leq 1000$) beriladi.

Chiqish: Aylanalar kesishsa “YES”, aks holda “NO” degan yozuvni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 0 2 0 3 2	YES
2	1 1 1 4 4 1	NO

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int x,y,y1,x1,r,r1;
x=sc.nextInt();
y=sc.nextInt(); // (x,y), 1- aylana koordinatalari.
r=sc.nextInt(); // r, 1- aylana radiusi.
x1=sc.nextInt();
y1=sc.nextInt(); // (x1,y1), 2- aylana koordinatalari.
r1=sc.nextInt(); // r1, 2- aylana radiusi.
int d=(int)Math.sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
// d, ikki aylana orasidagi masofa.
if((d<=r+r1)&&(r<=r1+d)&&(r1<=r+d)) pw.print("YES");
else pw.print("NO");
```

Tartib raqami №28

Simmetriya

Tekislikda L to'g'ri chiziq va A nuqta berilgan. B nuqta A nuqtaga L to'g'ri chiziqqa nisbatan simmetrik deyiladi, agar AB kesma L to'g'ri chiziqqa perpendikulyar bo'lsa va L to'g'ri chiziq bilan kesishish nuqtasida teng 2 ga bo'linsa. Xususiyl holda, agar A nuqta L to'g'ri chiziqda yotsa, B nuqta A nuqta bilan ustma-ust tushadi.

L to'g'ri chiziq koordinata o'qlaridan biriga parallel ravishda va A nuqta berilgan.

A nuqtaga L to'g'ri chiziqqa nisbatan simmetrik bo'lgan B nuqtani toping.

Kiritish: Birinchi qatorda 4 ta son: x_1, y_1, x_2, y_2 ikkita nuqta koordinatalari bo'lib, L to'g'ri chiziq shu nuqtalardan o'tadi. Ikkinchi qatorda 2 ta son- A nuqtaning koordinatalari.

Barcha sonlar modul jihatdan 10^8 dan katta emas.

Chiqish: B nuqtaning koordinatalari x_B va y_B ni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 0 0 1 10 10	-10 10
2	0 0 1 0 10 10	10 -10

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int x[]=new int[3] ;
int y[]=new int[3];
int z=0,q=0,i,j,x1=0,y1=0;
for(i=0;i<3;i++){ x[i]=sc.nextInt();y[i]=sc.nextInt();}
// (x[i],y[i])lar to'g'ri chiziq uchlar va A nuqta koordinatalari.
if(x[0]!=x[1]){ y1=2*y[0]-y[2]; x1=x[2];}
/* Agar to'g'ri chiziqning x o'qi koordinatalari bir biriga teng bo'lmasa u holda A
nuqtaga simmetrik nuqtaning x koordinatasi A nuqtaning x koordinatasiga teng
bo'ladi. */
else if(y[0]!=y[1]) { x1=2*x[0]-x[2]; y1=y[2];}
pw.print(x1+" "+y1);
```

Tartib raqami №440

Biatlon

5 ta nishon berilgan. Ularning har biri 10 sm radiusli doira bo'lib, markazlari orasidagi masofa 25 sm. Markazlar 1 ta gorizontal chiziqda yotadi.

Koordinatalar sistemasini shunday quramizki, koordinatalar boshi eng chapdagi nishon markazida joylashgan bo'lib, OX o'ngga, OY yuqoriga yo'nalgan. Markazlari (0,0),...(100,0).

Nechta nishonlarga o'q tekkanini hisoblash kerak. Nishonga hech bo'lmasa birta o'q tekxa, bu nishon hisobga olinadi (nishonga 2 yoki ko'p marta tekxa ham bu birta nishon deb hisoblanadi).

Har bir sportchida 5 tadan o'q bor. Sportchining o'qlari tekkan nuqtalar koordinatalari beriladi. O'q tekkan nishonlar sonini toping.

Kiritish: 5 ta qator – har birida i-o'q tekkan nuqta koordinatalari x_i va y_i beriladi. Barcha sonlar moduli bo'yicha 1000 dan katta emas.

Chiqish: O'q tekkan nishonlar soni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 0 25 0 50 0 75 0 100 0	5
2	0 0 0 0 0 0 75 0 100 0	3

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int x[]=new int[5] ;
int y[]=new int[5];
int yy[]=new int[5];
int xx[]=new int[5];
int z=0,q=0,i,j;
```

```

double d=0;
for(i=0;i<5;i++){ x[i]=sc.nextInt();y[i]=sc.nextInt();}
for(i=0;i<5;i++){ xx[i]=i*25; yy[i]=0;}
for(i=0;i<5;i++){ z=0;
for(j=0;j<5;j++){
d= Math.sqrt((xx[i]-x[j])*(xx[i]-x[j])+(yy[i]-
y[j])*(yy[i]-y[j])); if(d<=10)z++; }
/* Shartda har bir o'q teggan nuqtaning koordinatalarini nishon koordinatalari
orasidagi farq topilyapti va shu farq nishon radiusidan kichik bo'lsa nishonga
nechanchi o'q tegganli z orqali hisoblab boriladi*/
if(z!=0)q++; }
/* z noldan farqli bo'lsa nishonga o'q teggan hisoblanadi va q orqali barcha o'q
teggan nishonlar soni hisoblab boriladi */
pw.print(q);

```

Tartib raqami №182

To'g'ri to'rtburchak-2

To'g'ri to'rtburchakning 3 ta uchi koordinatalari berilgan. 4-uchi koordinatalarini toping.

Kiritish: To'g'ri to'rtburchakning uchta uchining koordinatalari $x_1, y_1, x_2, y_2, x_3, y_3$ - orasida bo'sh joy bilan kiritilgan.

Chiqarish: To'rtinchi uchining koordinatalarini orasida bo'sh joy orqali chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 3 0 0 5 0	5 3
2	1 4 8 3 7 6	2 1

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

int x,x1,x2,x3,y,y1,y2,y3;
double s=0,s1=0,s2=0,s3=0,p=0,a,b,c,a1,b1,c1;
x =sc.nextInt(); y =sc.nextInt();

```

```

x1=sc.nextInt(); y1=sc.nextInt();
x2=sc.nextInt(); y2=sc.nextInt();
a=Math.sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
b=Math.sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
c=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
//a,b,c berilgan uchta tomonlar uzunliklari topilyapti.
if((a>b)&&(a>c))pw.print((x+x1-x2)+" "+(y+y1-y2)); else
if((b>a)&&(b>c))pw.print((x+x2-x1)+" "+(y+y2-y1)); else
pw.print((x1+x2-x)+" "+(y1+y2-y));

```

Tartib raqami №69

N burchakli muntazzam g'ildirak

Muntazam N –burchakli ko'pburchak shaklidagi g'ildirak ishlab chiqarilmoqda. Bu g'ildirak sifatli deyiladi, agar unga tashqi va ichki chizilgan aylana radiuslarining ayirmasi $(R-r) < 1$ bo'lsa. N son va N-burchak tomoni uzunligi A berilgan. Bu g'ildirak sifatli yoki sifatsiz ekanligini aniqlang.

Kiritish: N va A ($3 \leq N \leq 1000$, $1 \leq A \leq 1000$) orasida bo'sh joy bilan kiritilgan.

Chiqarish: Sifatli bo'lsa – “YES” aks holda - “NO”

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	3 1	YES
2	239 566	NO

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

int n=sc.nextInt();
String s=sc.next();
double x,a,t,r;
a=Double.parseDouble(s);
if(n>3){ t=(180-(double)(360*1.0/n))/2;
// t, N burchakning burchakning radian qiymati.

```

```

x=Math.PI*t/180;
// x, N burchakning burchakning radian qiymati.
r=a*(1-Math.sin(x))/(2*Math.cos(x));
    } else r=a*1.0/(2*Math.sqrt(3));
// r, radiuslar ayirmasi
if(r<1) pw.print("YES"); else pw.print("NO");

```

Tartib raqami №55

Fonarlarni

Misha 1 ta va 2 ta lampochkali fonarlarni devorga tutib, qaysinisi yaxshiroq yoritishini aniqlamoqchi. Fonarlarni hosil qilgan doiralarni koordinatalari va radiuslari aniqlangan (radiuslari teng). Shuningdek, 1 ta lampochka uchun yoritilgan yuzani S ma'lum.

Kiritish: Dastlabki ikkita qatorda doiralarni markazlari koordinatalari (x_1, y_1) va (x_2, y_2) berilgan. Uchinchi qatorda esa qatorda r radius, to'rtinchi qatorda esa fonarlarni yoritayotgan doira yuzasi s berilgan. Barcha sonlar butun va quyidagicha chegaralangan: $1 \leq x_1, y_1, x_2, y_2, r \leq 100$, $1 \leq s \leq 10^5$. Har xil fonarlarni yorituvchi yuzalar bir-biridan 10^{-3} dan kattaroq songa farq qiladi.

Chiqarish: Ikkita lampochkali fonar yaxshiroq yoritinsa – “YES”, aks holda – “NO”.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	1 2 3 4 2 22	YES
2	1 1 100 100 1 7	NO

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int x,y,x1,y1; x=sc.nextInt(); y=sc.nextInt();
x1=sc.nextInt(); y1=sc.nextInt();
String s=sc.next(),s1=sc.next();
double r,t,d=0,t1=0,f,h; r=Double.parseDouble(s);
//r, fonar yoritadigan doira yuza radiusi.
t=Double.parseDouble(s1);
//t, bitta fonar yoritadigan yuza.
d=Math.sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
/* d, fonarlar yoritadigan doiralar markazlari orasidagi masofa. */
if((d<2*r)&&(d!=0)){
h=(double)Math.sqrt(r*r-d*d/4);
/* h, doiralar kesishish nuqtasidan ular markazlarini tutashtiruvchi to'g'ri chiziqqa
tushirilgan balandlik uzunligi. */
f=r-d/2;//radius va oraliq masofa yarmi orasidagi farq.
t1=2*r*r*Math.PI-8*f*h/3; }
//t1, ikkita fonar orqali yoritiladigan yuza.
else if((d==2*r)|| (d!=0)) t1=2*r*r*Math.PI;
else t1=r*r*Math.PI;
if(t1>t) pw.print("YES"); else pw.print("NO");
```

Tartib raqami №102

Uchburchak

Dekart koordinatalar sistemasida uchburchakning uchlari va bitta nuqtaning koordinatalari berilgan. Shu berilgan nuqta uchburchakda yotadimi?

Kiritish: Birinchi uchta qatorda uchburchakning koordinatalari (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , to'rtinchi qatorda nuqtaning koordinatalari (x_4, y_4) . Barcha koordinatalar qiymatlari 10000 dan oshmaydi.

Chiqarish: Nuqta uchburchak ichida yotsa -“In”, aks holda “Out” degan yozuvni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 0 100 0 0 100 100 100	Out
2	0 0 100 0 0 100 10 10	In
3	0 0 100 0 0 100 50 50	In
4	0 0 100 0 0 100 0 0	In

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int x,x1,x2,x3,y,y1,y2,y3;
double s=0,s1=0,s2=0,s3=0,p=0,a,b,c,a1,b1,c1;
x=sc.nextInt(); y=sc.nextInt();
x1=sc.nextInt(); y1=sc.nextInt();
x2=sc.nextInt(); y2=sc.nextInt();
x3=sc.nextInt(); y3=sc.nextInt();
a=Math.sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
b=Math.sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
c=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
p=(a+b+c)*1.0/2;
s=Math.sqrt(p*(p-a)*(p-b)*(p-c));
```

```

a1=Math.sqrt((x3-x1)*(x3-x1)+(y3-y1)*(y3-y1));
b1=Math.sqrt((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));
c1=Math.sqrt((x3-x)*(x3-x)+(y3-y)*(y3-y));
p=(a1+b1+c)*1.0/2;
s1=Math.sqrt(p*(p-a1)*(p-b1)*(p-c));
p=(b1+b+c1)*1.0/2;
s2=Math.sqrt(p*(p-b1)*(p-b)*(p-c1));
p=(a+a1+c1)*1.0/2;
s3=Math.sqrt(p*(p-a)*(p-a1)*(p-c1));
/* Nuqta bilan uchburchakning ikkitadan uchini olib uchburchakni uchta
uchburchakka ajratiladi va shu uchburchaklarning yuzalari yig'indisi
uchburchakning yuzidan 0.0001 ga farq qilsa, nuqta shu uchburchak ichida yotadi.
*/
if((Math.abs(s-(s1+s2+s3))>=0)
&&(Math.abs(s-(s1+s2+s3))<=0.0001))pw.print("In");
else pw.print("Out");

```

Tartib raqami №90

Uchburchakli mamlaklar

N shahar koordinatalari x_n, y_n . Bu tekis planetadagi davlatlar esa uchburchak shaklga ega. Aniqlash kerak – qaysi davlatlarga N tegishli bo'lgan? (Aniqrog'i, N nuqta berilgan uchburchakdan qaysilarining ichida yotishini aniqlang).

Kiritish: Birinchi qatorda N ning koordinatalari $-x_n, y_n$. Ikkinchi qatorda k –uchburchaklar soni. Keyingi k ta qatorda 6 ta butun sonlar –uchburchak uchlarining koordinatalari $(x1;y1), \dots, (x3;y3)$ berilgan. Birorta mamlakatlatning maydoni nolga teng emas. Barcha koordinatalar qiynatlari 10000 dan katta emas.

Chiqarish: Birinchi qatorda N tegishli bo'lgan davlatlar soni.

Ikkinch qatorda o'sish tartibida bu davlatlar(uchburchaklar) nomeri chiqarilsin.

Uchburchaklar kirish faylidagi nomeri bo'yicha nomerlanadi.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 1 2 -2 0 2 0 0 2 -3 0 3 0 0 3	2 1 2
2	0 2 2 -2 0 2 0 0 2 -3 0 3 0 0 3	1 2

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

int x,x1,x2,x3,y,y1,y2,y3,j=0,w=0,i,n;
double s=0,s1=0,s2=0,s3=0,p=0,a,b,c,a1,b1,c1,q=0;
x3=sc.nextInt(); y3=sc.nextInt(); n=sc.nextInt();
int aa[]=new int [n];
for(i=0;i<n;i++){ q=0;
x=sc.nextInt(); y=sc.nextInt();
x1=sc.nextInt(); y1=sc.nextInt();
x2=sc.nextInt(); y2=sc.nextInt();
a=Math.sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
b=Math.sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
c=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
p=(a+b+c)*1.0/2; s=Math.sqrt(p*(p-a)*(p-b)*(p-c));
a1=Math.sqrt((x3-x1)*(x3-x1)+(y3-y1)*(y3-y1));
b1=Math.sqrt((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));
c1=Math.sqrt((x3-x)*(x3-x)+(y3-y)*(y3-y));
if((a1+b1>c)&&(Math.abs(a1-b1)<c)){ p=(a1+b1+c)*1.0/2 ;
s1=Math.sqrt(p*(p-a1)*(p-b1)*(p-c)); } else q++;
if((c1+b1>b)&&(Math.abs(c1-b1)<b)){ p=(b1+b+c1)*1.0/2 ;
s2=Math.sqrt(p*(p-b1)*(p-b)*(p-c1)); }
if((a1+c1>a)&&(Math.abs(a1-c1)<a)){

```

```

p=(a+a1+c1)*1.0/2 ;
s3=Math.sqrt (p*(p-a)*(p-a1)*(p-c1)); } else q++;
if ((q==0) && (Math.abs (s-(s1+s2+s3))>=0) && (Math.abs (s-
(s1+s2+s3))<=0.0001)) { aa[j]=i+1; j++; w++; } }
//Har bir uchburchak uchun №102 masalaga o'xshash bajariladi.
pw.println(w);
for(i=0;i<j;i++)pw.print(aa[i]+" ");

```

Tartib raqami №390

Uchburchak soha

Uchburchak uchlari koordinatalari va uni ichida yotuvchi nuqta x_c, y_c koordinatalari berilgan. Bu nuqtadan uchburchak tomonlarigacha bo'lgan masofalar ichida eng qisqasini toping.

Kiritish: Birinchi qatorda 6 ta son uchburchak uchlari koordinatalari $x_1, y_1, x_2, y_2, x_3, y_3$, ikkinchi qatorda nuqta koordinatalari x_c, y_c berilgan. Uchburchak o'tmas burchakli emas va barcha koordinatalar qiymatlari 10000 dan oshmaydi.

Chiqarish: Kerakli masofani 10^{-6} dan kichik bo'lmagan aniqlikda chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	0 0 1 2 2 0 1 1	0.4472135
2	0 0 0 3 3 0 1 1	0.7071067

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

import java.util.*;
import java.io.*;
class tch {
double asz( double x, double y, double x1, double y1, d
ouble x2, double y2){

```

```

double a,b,c;
a=y1-y2; b=x2-x1; c=y2*x1-y1*x2;
return Math.abs((a*x+b*y+c)/(Math.sqrt(a*a+b*b))); } }
/* tch degan sinf va bu sinfga tegishli asz medodi yaratilyapti. Bu metod natija
sifatida haqiqiy sonni qaytaradi va berilgan nuqtalarga mos nuqtadan to'g'ri
chiziqqacha bo'lgan eng qisqa masofani ya'ni perpendikulyar uzunligini
hisoblaydi. */
class Main { public static void main(String taxmin[])
throws IOException{
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
double x0,x1,y0,y1,x2,y2,x,y,l1,l2,l3;
x0=Double.parseDouble(sc.next());
y0=Double.parseDouble(sc.next());
x1=Double.parseDouble(sc.next());
y1=Double.parseDouble(sc.next());
x2=Double.parseDouble(sc.next());
y2=Double.parseDouble(sc.next());
x=Double.parseDouble(sc.next());
y=Double.parseDouble(sc.next());
tch ga=new tch ();
l1=ga.asz(x,y,x1,y1,x0,y0);
l2=ga.asz(x,y,x1,y1,x2,y2);
l3=ga.asz(x,y,x2,y2,x0,y0);
if(l1<l2) l2=l1; if(l3<l2) l2=l3; pw.print(l2);
sc.close(); pw.close(); } }

```

2.4. Rekursiya va o'rin almashtirishlar (Рекурсия, перебор).

Tartib raqami №352

Kasr

Vali beshinchi sinfda o'qiydi va hozirda ular "Naturl sur'at va maxrajli oddiy kasrlar" mavzusiga kelishgan. Oddiy kasr to'g'ri qisqarmaydigan kasr deyiladi, kasrning sur'ati maxrajidan kichik va o'zaro tub sonlar bo'lsa. Vali matematikani juda yaxshi ko'radi va o'zi mustaqil ko'plab masalalarni yechadi. Ayni vaqtda Vali sur'at va maxrajining yig'indisi N ga teng bo'lgan eng katta qisqarmaydigan to'g'ri kasrni qidirayпти.

Valiga bu masalani yechishda yordam beruvchi dastur tuzing.

Kiritish: N soni kiritiladi ($3 \leq N \leq 2 \cdot 10^9$).

Chiqarish: Izlanayotgan kasr sur'at va maxrajini bo'sh joy bilan ajratilgan holda chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	3	1 2
2	10	3 7

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int i,a,b,j,n=sc.nextInt(); i=n/2+1;
double x; a=i; b=n-i;
while(i<n){ j=2; a=i; b=n-i;
if((a%j==0)&&(b%j==0)){ a=a/j; b=b/j; } j=3;
x=Math.sqrt(n-i);
while(j<=x){ if((a%j==0)&&(b%j==0)){a=a/j;b=b/j; break;}
else j=j+2; } if(a+b==n)break; i++; }
pw.print(b+" "+a);
```

Tartib raqami №171

Bo'luvchilari soni

X natural soni beriladi. Uning bo'luvchisi deyiladi, agar $1 \leq y \leq x$ x ni y ga bo'lganda qoldiq nolga teng bo'lsa.

X ning barcha bo'luvchilari sonini toping.

Kiritish: x soni kiritiladi $x(1 \leq x \leq 10^{18})$. x ning barcha tub bo'luvchilari 1000 dan ortiq emas.

Chiqarish: Masala shartiga mos javobni chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	12	6
2	239	2

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

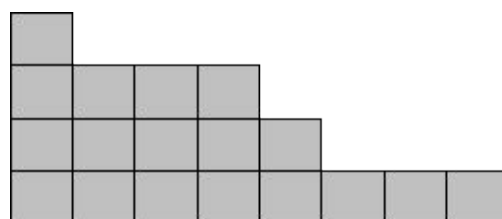
long a=Long.parseLong(sc.next());
long q=1,w=0,i=2,j=0,t=a,f; int h=0;
long aa[]=new long[10000];
while(h<10000) { f=2;
while(f<=Math.sqrt(i)) {
if((i%f==0)) {w=1; break; } f++; }
if(w==0){aa[h]=i; h++;}w=0; i++; } h=0;
while((aa[h]<=Math.sqrt(t)) && (a!=1)) {
if((a%aa[h]==0)) {a=a/aa[h]; w=1; j++;}
else { if(w==1)q=q*(j+1); j=0; w=0; h++;} }
if(w==1)q=q*(j+1); if(t==1) pw.print(1); else
if(q==1)pw.print(2); else pw.print(q);

```

Tartib raqami №16

Zina poya

Zina poya deb shunday kubiklar to'plamiga aytiladiki, har bir yuqori qatlami, quyi qatlamdan kam kubik saqlaydi. Shunday dastur tuzingki u berilgan N ta kubdan nechta zina poya tayyorlash mumkin.



Kiritish: Kublar soni N kiritilgan ($1 \leq N \leq 100$).

Chiqarish: Berilgan n ta kubdan nechta zina poya qurish mumkin.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	3	2
2	6	4

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

yechim 1

```
import java.util.*;
import java.io.*;

class Main {      public static long ACount(byte n,byte m) {
long f; f=(byte) 1; if(n!=0) f--; while (m<n) { m++;
f+=ACount((byte) (n-m),m); } return f; }

public static void main(String[] args) throws IOException {
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));

byte n = sc.nextByte();
pw.println(ACount((byte)n,(byte)0));
pw.close(); } }
```

yechim 2

```
import java.util.*;
import java.io.*;

class Main { static long N,res=0;

public static void ARec(long a,long b) { long i;
if (a==0) res++; else { for(i=1;i<=a;i++) if (i<b) ARec(a-i, i);
else break; } }

public static void main(String[] args) throws IOException {
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));

N = sc.nextByte(); ARec(N, N+1); pw.println(res); pw.close(); } }
```

2.5. Murakkab algoritmlar (Жадный алгоритм).

Tartib raqami №398

Yig'indi-2

x natural son berilgan. Bu sonni 4 ta natural sonlar yig'indisi ko'rinishida ifodalash usullari sonini aniqlang, ya'ni $x = a + b + c + d$ kabi, bu yerda $a \leq b \leq c \leq d$.

Kiritish: $x(1 \leq x \leq 1500)$ natural son kiritilgan.

Chiqarish: Masalaning javobini chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	3	0
2	5	1

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
int i, l=0, j=0, k=0, x; long w=0; x=sc.nextInt();
for(i=1; i<=x/4; i++) { if(i+i+i+i>x) break;
/* Agar (i+i+i+i>x) bolsa u holda x ni a+b+c+d ko'rinishda ifodalab bolmaydi
chunki i<=l<=k<=d shuning uchun takrorlash sindirilgan. Qalgan hollarda ham
takrorlash shu sababdan sindiriladi.*/
for(l=i; l<=x/3; l++) { if(l+l+l+i>x) break;
for(k=l; k<=x/2; k++) { if(k+l+i+k>x) break;
if(x-(k+l+i)>=k) w++; } } }
/* (x-(k+l+i)>=k) bunda ixtiyoriy k dan kichik bo'lmagan d sonni qo'shish orqali x
ni a+b+c+d ko'rinishda ifodalash mumkin. Masalani yechish davomida sikllardagi
qisqartirishlar(sindirishlar) dastur ishlash vaqti bir sekuntдан oshmasligiga ham
yordam beradi. */
pw.println(w);
```

Tartib raqami №385

O'lchagich bilan ishlash

Bir nechta nuqtalar o'z koordinatalari bilan berilgan. Shu nuqtalardan juft-juft olinib, orasidagi masofalar hisoblangan. Ushbu masofalar orasida nechta xil uzunlikdagi masofalar uchrashini toping.

Kiritish: 1-qatorda n – nuqtalar soni. Keyingi n ta qatorda 2 tadan son - mos nuqtalar koordinatalari berilgan. Koordinata qiymatlarning absolyut kattaligi 10000 dan oshmaydi.

Chiqarish: 1- qatorda k – turli masofalar soni. Keyingi k ta qatorda haqiqiy sonlar – masofaning o'zlari chiqarilsin. Masofalar o'sish tartibida chiqarilsin. Sonlarning aniqligi 10^{-9} dan kichik bo'lmasligi kerak.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	4 0 0 1 1 1 0 0 1	2 1.0 1.414213562373

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```

TreeSet a=new TreeSet ();
int n=sc.nextInt();
int b[]=new int[n];
int b1[]=new int[n];
int i,j;
for(i=0;i<n;i++){b[i]=sc.nextInt();b1[i]=sc.nextInt();}
for(i=0;i<n;i++)
for(j=i+1;j<n;j++){
a.add(Math.sqrt((b[i]-b[j])*(b[i]-b[j])+(b1[i]-b1[j])*(b1[i]-b1[j]))));
} double ss=Double.parseDouble("1.0");
Iterator vv=a.iterator();

```

```

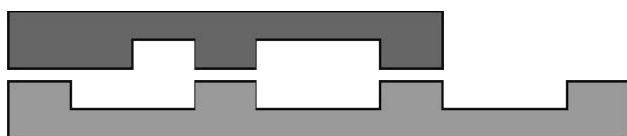
pw.println(a.size());
while(vv.hasNext()){
String  ass=(String) (" "+vv.next());
pw.println(ass); }

```

Tartib raqami № 269

Tormoz mexanizmi

Yangi tormoz mexanizmi ishlab chiqildi. Bu mexanizm 2 ta bo'lakdan tuzilgan bo'lib, ular bir-biriga mahkamlanadi(kiydiriladi). Har bir bo'lak n uzunlikka ega bo'lib, n ta qismga bo'lingan, bu qismlarning har biri h yoki $2h$ balandlikka ega. Shunday qilib, bo'laklar tishli shakl ko'rinishiga egadir.



O'zaro ta'sir vaqtida 2 ta bo'lak bir-biriga kiydiriladi, bunda birlashmaning umumiy uzunligi eng kichik bo'lishi kerak.



Umumiy konstruksiya balandligi $3h$ dan oshmasligi kerak.
Bo'laklarni aylantirish yoki tishlarni olib tashlash man etiladi.

Kiritish: ikkita satr – har birida bittadan bo'lak konfiguratsiyasi(tuzilishi) berilgan. Konfiguratsiya birinchi va ikkinchi ketma-ketligidan iborat bo'lib, mos tishcha balandligi (h yoki $2h$) ni bildiradi. Hech bir satr bo'sh emas, satr uzunligi 100 dan oshmaydi.

Chiqarish: Berilgan bo'laklardan tuzilgan eng kichik uzunlikdagi konstruksiya uzunligini chiqaring.

Masalan:

№	INPUT.TXT	OUTPUT.TXT
1	2112112112	10

	2212112	
2	12121212 21212121	8
3	2211221122 21212	15

Bu masala yechimining *java* dasturlash tilida ifodalanishi quyidagicha:

```
String s,s1; int i,n,m,p=0,x,y,r,r1=0,j;
s=sc.next(); s1=sc.next(); n=s.length();
m=s1.length(); x=m;
boolean t=true;
for(i=0;i<n;i++){s1="0"+s1; s1=s1+"0"; }
m=s1.length(); r=m;
for(i=0;i<m-n;i++){
for(j=0;j<n;j++){ p=0;
p=p+Integer.parseInt(""+s.charAt(j))+Integer.parseInt("
"+s1.charAt(j+i));
if(p>3) {t=false; break;} else t=true; } r1=0;
if(t) { if(i-n>0)r1=r1+i-n;
if((x+n)-(i+n)>0)r1=r1+(x+n)-(i+n); r1=r1+n;
if(r1<r){r=r1; } } } pw.print(r);
```

Foydalanilgan adabiyotlar ro'yxati.

1. Ўзбекистон Республикаси “Кадр тайёрлаш миллий дастури” // Баркамол авлод.-Ўзбекистон тараккиётининг пойдевори.-Т.: “Шарк” нашриёт-матбаа концерни,1997.-Б.31-61
2. Каримов И.А. “Юксак маънавият енгилмас куч”. –Т.: маънавият,2008-176 б.
3. Бақоев О, Болтаев Т. Б “ JAVA тилида тузилган дастурлар” Бухоро – 2009, 92 б.
4. Бақоев О, Болтаев Т. Б “ JAVA тилида ДАСТУРЛАШ” Бухоро – 2009, 64 б.
5. Герберт Шилдт “Полный справочник по Java™ “ –М.: Издательский дом “ Вильямс”–2007,1024 б.
6. [www//acmp.ru](http://acmp.ru)
7. [www//dasturchi.uz](http://dasturchi.uz)

Mundarija.

I. Internet tizimida olimpiada masalalari va ularni yechish.	5
1.1 Dasturlash bo'yicha olimpiadaga oid saytlar.....	5
1.2. Java dasturlash tilida olimpiada masalalarini echish va tekshirish.	7
II. Mavzulashtirilgan olimpiada masalalari.....	19
2.1. Boshlovchilar uchun (Начинающих).....	19
2.2. Butun sonlar arifmetikasi (Целочисленное).	33
2.3. Geometriya (Геометрия).	37
2.4. Rekursiya va o'rin almashtirishlar (Рекурсия, перебор).....	49
2.5. Murakkab algoritmlar (Жадный алгоритм).....	53
Foydalanilgan adabiyotlar ro'uxati.	57

[illegible]