

Юнусова М.Д., Раҳимов А.Б.

Дастурлаш бўйича практикум

**Касб-ҳунар коллеж талабалари учун
ўқув қўлланма**

МУНДАРИЖА

СЎЗ БОШИ	4
I ҚИСМ. НАЗАРИЙ МАЪЛУМОТЛАР	6
§ 1.1. Turbo Pascalда дастур тузиш учун керакли маълумотлар	6
1.1.1. Дастур таснифи.....	6
1.1.2. Маълумотларнинг асосий турлари	6
1.1.3. Массивлар	7
1.1.4. Ёзувлар	7
1.1.4. Кўрсатмалар (операторлар).....	8
1.1.5. Функцияларни эълон қилиш	11
1.1.6. Процедураларни эълон қилиш	11
§ 1.2. Turbo Pascal резерв сўзлари, директивалари, стандарт функция ва процедуралари.....	12
1.2.1. Резерв сўзлар ва директивалар.....	12
1.2.2. Turbo Pascal директивалари	12
1.2.3. Функция ва процедуралар	12
1.2.3.1. Математик функциялар.....	14
1.2.3.2. Ўзгартиришнинг функция ва процедуралари	15
1.2.3.3. Сатр ва белгилар билан ишлаш учун функциялар ва процедуралар.....	16
1.2.3.4. Файллар билан ишлаш учун процедура ва функциялар	17
1.2.3.5. Бошқа функциялар.....	19
§ 1.3. Ўзгарувчиларнинг хотирада сақланиш соҳаси.....	22
§ 1.4. ЭҲМда паскал тилида ёзилган дастурлар билан ишлаш тартиби ..	24
II ҚИСМ. МАСАЛАЛАР ШАРТЛАРИ.....	30
§ 2.1. Қуйи мураккабликдаги масалалар	30
2.1.1. Ўзгарувчиларни эълон қилиш.....	30
2.1.2. Ўзлаштириш кўрсатмаси	31
2.1.3. Чиқариш кўрсатмаси.....	31
2.1.4. Киритиш кўрсатмаси.....	33
2.1.5. Чизиқли таснифли дастурлар	34
2.1.6. Танлов (If) кўрсатмаси.....	38
2.1.7. CASE кўрсатмаси	42

2.1.8. FOR кўрсатмаси.....	43
2.1.9. REPEAT кўрсатмаси	49
2.1.10. WHILE кўрсатмаси.....	51
2.1.11. Белгилар ва қаторлар	52
2.1.12. Массивлар	54
2.1.13. Функциялар.....	61
2.1.14. Процедуралар.....	63
2.1.15. Файллар	65
2.1.16. Рекурсия	70
§ 2.2. Юқори мураккабликдаги масалалар	72
III ҚИСМ. МАСАЛАЛАР ДАСТУРЛАРИ	80
§ 3.1. Қуйи мураккабликдаги масалалар дастурлари.....	80
3.1.1. Чиқариш кўрсатмали дастурлар	80
3.1.2. Чизиқли таснифли дастурлар	81
3.1.3. IF танлов кўрсатмали дастурлар	84
3.1.4. CASE кўрсатмали дастурлар.....	94
3.1.5. FOR кўрсатмали дастурлар	97
3.1.6. REPEAT кўрсатмали дастурлар.....	110
3.1.7. WHILE кўрсатмали дастурлар	112
3.1.8. Белгилар ва қаторлар ишлатилган дастурлар.....	114
3.1.9. Массивларга оид дастурлар	124
3.1.10. Функциялар иштирокидаги дастурлар.....	146
3.1.11. Процедурали дастурлар	153
3.1.12. Файлларга оид дастурлар	155
3.1.13. Рекурсияли дастурлар.....	165
§ 3.2. Юқори мураккабликдаги масалалар алгоритмлари ва дастурлари	171
Фойдаланилган адабиётлар рўйхати	243

СЎЗ БОШИ

Замонавий ЭҲМлар (компьютерлар) ўзларининг аввалги авлодлари каби ишлаш чоғида энг оддий арифметик ва мантикий амалларни бажара олади, холос. Машина мураккаб масалани ечиши учун унинг хотирасига мос дастур киритилган бўлиши керак. Дастур буйруқлар кетма-кетлигидан, кўрсатмалар – бирор оддий амални бажаришга оид машинага бериладиган кўрсатмалардан иборат бўлиб, анча мураккаб амални ифодалайди.

Мазкур китоб колледж талабаларида, шунингдек, дастурлаш билан қизиқувчиларда дастурлаш бўйича бошланғич кўникмалар ҳосил қилишга бағишланган.

ЭҲМларга дастур-топшириқ ёзиш учун исталган дастурлаш тилини танлаб олиш мумкин. Китобда шундай тил сифатида деярли барча дастурлаш тилларида учрайдиган тушунча ва тузилишларни ўз ичига олган Турбо Паскал тили танлаб олинган. Бу эса китобхонга кейинчалик бошқа тиллар билан танишиш чоғида дастурлаш соҳасидаги билимларни кенгайтиришга қулай имкон беради.

Дастурлашни ўрганиш учун дастурлаш тилига бағишланган китобни ўқиб чиқиш етарли эмас, бунинг учун масалаларнинг дастурларини ёзиш керак.

Бу китобда бошловчи дастурчи учун у еча олиши мумкин бўлган масалалар тўплами келтирилган.

Китоб 3 қисмдан иборат.

I қисмда Турбо Паскал дастурлаш тили бўйича, дастурлашда асқотадиган энг зарурий тушунчаларни ўз ичига олган ёрдам маълумотномаси келтирилган.

II қисмда масалалар шартлари шартлари берилган. Улар қуйи ва юқори мураккабликдаги масалаларга ажратилган. Қуйи мураккабликдаги масалалар мавзуларига кўра гуруҳланган. Юқори мураккабликдаги масалалар эса мавзусига эмас, мураккаблигига кўра бирлаштирилган. Улар орасида ҳар хил мавзуларга тегишлилари бўлиб, улар асосан иқтидорли талабаларга мўлжаллаб берилган.

III қисмда масалалар асосий қисмининг дастур матнлари келтирилган, уларни ўрганиш, дастурчига, агар у масалани мустақил ечган бўлса ҳам, фойда келтириши, шубҳасиз.

Юқори мураккабликдаги масалаларнинг эса фақат ечиш дастурларигина эмас, балки уларни ечиш алгоритмлари ҳам баён этилган. Улар дастурларни ўқишга ёрдам берадиган даражада ишлаб чиқилган.

Масалани ечишга ўтишдан олдин талаба тегишли мавзунини ўрганиши - дарсликнинг бўлимини ўқиб чиқиши керак. Агар масала ечими дарров топилмаса, унинг ечимини кўриш ва яна бир марта масалани мустақил ечишга урнаши мумкин.

Масала шартида айтилганидек, дастур ишласа, демак, масала ечилган ҳисобланади.

I ҚИСМ. НАЗАРИЙ МАЪЛУМОТЛАР

§1.1. TURBO PASCAL ДА ДАСТУР ТУЗИШ УЧУН КЕРАКЛИ МАЪЛУМОТЛАР

1.1.1. ДАСТУР ТАСНИФИ

Pascal тилидаги дастур куйидаги бўлимлардан иборат:

- Нишона (метка) ларни эълон қилиш бўлими;
- Ўзгармасларни эълон қилиш бўлими;
- Ўзгарувчиларни эълон қилиш бўлими;
- Процедура ва функцияларни эълон қилиш бўлими;
- Дастур кўрсатмалари бўлими

Умумий ҳолда дастур таснифи куйидаги кўринишга эга:

```
label      {меткаларни эълон қилиш}
const     {ўзгармасларни эълон қилиш}
type      {турларни эълон қилиш}
var       {ўзгарувчиларни эълон қилиш}
         {дастурчининг процедура ва функцияларини эълон қилиш}
begin
         {асосий дастур кўрсатмалари}
end.
```

1.1.2. МАЪЛУМОТЛАРНИНГ АСОСИЙ ТУРЛАРИ

Pascal тилининг асосий турларига куйидагилар киради:

- бутун сонлар (INTEGER ва бошқалар);
- ҳақиқий сонлар (REAL ва бошқалар);
- белгилар (CHAR);
- сатрлар (STRING);
- мантиқий (BOOLEAN).

Бутун ва сузувчи нуқтали сонлар ҳар хил форматларда берилиши мумкин.

Бутун сонлар

Формат	Қийматлар соҳаси
SHORTING	- 128 – 127
INTEGER	- 32768 – 32767
LONGINT	- 2147483648 - 2147483647

BYTE	0 – 255
WORD	0 – 65535

Ҳақиқий сонлар

Формат	Қийматлар соҳаси	Рақамлар қиймати миқдори
REAL	2.9E – 39 – 1.7e 38	11 – 12
SINGLE	1.5E – 45 – 3.4e 38	7 – 8
DOUBLE	5.0E – 324 – 1.7e 108	15 – 16
EXTENDED	3.4E – 4932 – 1.1e 4932	–19 – 20

Сатрлар

Сатрлар икки хил кўринишда эълон қилиниши мумкин:

1-кўриниш:

Исм : String;

255 белгидан иборат узунликка эга ўзгарувчи эълон қилинади.

2-кўриниш:

Исм: String [сатр узунлиги];

Кўрсатилган узунликдаги ўзгарувчи эълон қилинади.

1.1.3. МАССИВЛАР

Бир ўлчовли массивни эълон қилиш:

Массив исми: агау [қуйи индекс .. юқори индекс] of <элементлар тури>;

Икки ўлчовли массивни эълон қилиш:

Массив исми: [қуйи индекс 1 .. юқори индекс 1, қуйи индекс 2 .. юқори индекс 2] of <элементлар тури>;

1.1.4. ЁЗУВЛАР

Ёзувларни икки хил кўринишда эълон қилиш мумкин:

1-кўриниш:

Ўзгарувчи Ёзув: **record**

Майдон 1 : 1 тур;

Майдон 2 : 2 тур;

....

Майдон 3 : 3 тур;

end.

2-кўриниш:

Аввал ёзув тури, кейин ўзгарувчи тур эълон қилинади:

type

Тур Ёзуви Исми: **record**

Майдон 1 : тур 1;

Майдон 2 : тур 2;

.....

Майдон *J* : тур *J*;

end;

var

Ўзгарувчи Ёзув : Тур Ёзуви Исми;

1.1.4. КЎРСАТМАЛАР (операторлар)

ТАНЛАШ кўрсатмалари

1) IF кўрсатмаси

1-кўриниш:

if шарт

then

begin

{бу кўрсатмалар, агар шарт тўғри бўлса, бажарилади}

end

else

begin

{бу кўрсатмалар шарт хато бўлса, бажарилади}

end;

2-кўриниш:

IF шарт

then

begin

{бу кўрсатмалар шарт тўғри бўлганда бажарилади}

end

Изоҳ: Агар **begin** ва **end** ўртасида фақат битта кўрсатма (оператор) бўлса, **begin** ва **end** ни ёзмаслик мумкин.

2) CASE кўрсатмаси

1-кўриниш:

Case ифода **of**
begin

```
    1 – ўзгармаслар рўйхати: begin
                                   {1-кўрсатмалар}
                                   end;
    2 – ўзгармаслар рўйхати: begin
                                   {2-кўрсатмалар}
                                   end;
    ...
    J – ўзгармаслар рўйхати: begin
                                   {J - кўрсатмалар}
                                   end;
else
                                   begin
                                   {кўрсатмалар}
                                   end;
end
```

Case сўзидан кейин келувчи ифода қиймати, тегишли рўйхат ўзгармаси билан мос келгандагина **begin** ва **end** ўртасидаги кўрсатмалар бажарилади, акс ҳолда **else** сўзидан кейин келувчи **begin** ва **end** сўзлари ўртасидаги кўрсатмалар бажарилади.

ЦИКЛАР

3) FOR кўрсатмаси

1-кўриниш: (ҳисобчи қиймати ўсади);

```
for Ҳисобчи: = Бошланғич Қиймат to Сўнгги Қиймат do
    begin
        {кўрсатмалар}
    end;
```

begin ва **end** ўртасидаги кўрсатмалар [(Сўнгги Қиймат – Бошланғич Қиймат) + 1] марта бажарилади.

Агар бошланғич қиймат сўнгги қийматдан катта бўлса, **begin** ва **end** ўртасидаги кўрсатмалар бажарилмайди.

Изоҳ: Агар **begin** ва **end** ўртасида фақат битта кўрсатма бўлса, **begin** ва **end** сўзларини ёзмаслик мумкин.

2-кўриниш (ҳисобчи қиймати камаяди)

```
for Ҳисобчи: = Бошланғич Қиймат downto Сўнгги Қиймат do  
    begin  
        {кўрсатмалар}  
    end;
```

begin ва **end** ўртасидаги кўрсатмалар [(Бошланғич Қиймат – Сўнгги Қиймат) + 1] марта бажарилади. Агар бошланғич қиймат сўнгги қийматдан кичик бўлса, **begin** ва **end** ўртасидаги кўрсатмалар бажарилмайди.

4) REPEAT кўрсатмаси

```
repeat  
    begin  
        {кўрсатмалар}  
    end  
until Шарт;
```

begin ва **end** ўртасидаги кўрсатмалар (цикл кўрсатмалари) бажарилади, шундан кейин Шарт ифоданинг қиймати текширилади. Агар у **False** га тенг бўлса (яъни шарт бажарилмаса), цикл кўрсатмаси яна бир марта бажарилади. Худди шундай тарзда Шарт ҳақиқат бўлгунча давом эттирилади. Шундай қилиб, **until** сўзидан кейин циклни тугатиш шarti ёзилади.

Изоҳ: Агар **begin** ва **end** ўртасида фақат битта кўрсатма ёзилган бўлса, **begin** ва **end** сўзларини ёзмаслик мумкин.

5) WHILE кўрсатмаси

```
while Шарт do  
    begin  
        {шартлар}  
    end;
```

Шарт ифодасидаги қиймат текширилади, агар у **True**га тенг бўлса (шарт бажарилса), **begin** ва **end** ўртасидаги кўрсатмалар (цикл кўрсатмалари) бажарилади. Шундан кейин яна Шарт ифодасининг қиймати текширилади, шундай тарзда Шарт ифодасининг қиймати **False**

бўлгунча давом эттирилади. Шундай қилиб, **while** сўзидан кейин цикл кўрсатмаларининг бажариш шарти ёзилади.

Изоҳ: Агар **begin** ва **end** ўртасида фақат битта кўрсатма ёзилган бўлса, **begin** ва **end** сўзларини ёзмаслик мумкин.

6) GOTO шартсиз ўтиш кўрсатмаси

GoTo метка (нишона)

Олдида, **label** бўлимида эълон қилинган, метка ёзилган кўрсатмага ўтиш амалга оширилади.

1.1.5. ФУНКЦИЯЛАРНИ ЭЪЛОН ҚИЛИШ

Function Функция Исми (**var** Параметр 1: Тур 1;

var Параметр 2: Тур 2;

...

var Параметр J : Тур J) : Тур;

const

{ўзгармасларни баён этиш}

var

{ўзгарувчиларни баён этиш}

begin

{функциялар кўрсатмалари}

Функция Исми := қиймати;

end;

1.1.6. ПРОЦЕДУРАЛАРНИ ЭЪЛОН ҚИЛИШ

Procedure Процедура Исми (**var** Параметр 1 : Тур 1;

var Параметр 2 : Тур 2 . . .

var Параметр J : Тур J);

const

{ўзгармасларни баён этиш}

var

{ўзгарувчиларни баён этиш}

begin

{процедура кўрсатмалари}

end;

§ 1.2. TURBO PASCAL РЕЗЕРВ СЎЗЛАРИ, ДИРЕКТИВАЛАРИ, СТАНДАРТ ФУНКЦИЯ ВА ПРОЦЕДУРАЛАРИ

1.2.1. РЕЗЕРВ СЎЗЛАР ВА ДИРЕКТИВАЛАР

Turbo Pascal тилининг резерв сўзлари

And	file	not	them
Array	for	object	to
Asm	function	of	type
Begin	go to	or	unit
Case	if	packed	until
Const	implementation	procedure	uses
Constructor	in	program	var
Destructor	inherited	record	while
Div	inline	repeat	with
Do	interface	set	xor
Downto	label	shl	
Else	mod	shr	
End	nil	string	

1.2.2. TURBO PASCAL ДИРЕКТИВАЛАРИ

Absolute	external	forward	near	public
Assembler	far	interrupt	private	virtual

1.2.3. ФУНКЦИЯ ВА ПРОЦЕДУРАЛАР

Математик	Изоҳ
ABS (<i>ифода</i>)	Аргументнинг абсолют қиймати (бутун ёки ҳақиқий тур)
SQR (<i>ифода</i>)	Аргумент квадрати (бутун ёки ҳақиқий тур)
SQRT (<i>ифода</i> : real) : real	Аргументдан квадрат илдиз чиқариш
SIN (<i>ифода</i> : real) : real	Раданларда ифодаланган аргументнинг синуси
COS (<i>ифода</i> : real) : real	Раданларда ифодаланган аргументнинг косинуси
ARCTAN (<i>ифода</i> : real) : real	Раданларда ифодаланган аргументнинг арктангенци
EXP (<i>ифода</i> : real) : real	Аргумент экспонентаси
LN (<i>ифода</i> : real) : real	Аргументнинг натурал логарифми

INT (<i>ифода</i> : real) : real	Аргумент бутун қисми
TRUNC (<i>ифода:real</i>) : longint	Аргумент бутун қисми
ROUND (<i>ифода:real</i>) : longint	Аргумент қийматини энг яқин бутун сонгача яхлитлаш
STR (<i>ифода: var Сатр</i> : string)	Сонли ифодани сатрга айлантириш
Val (<i>Сатр:string; var</i> Ўзгарувчи, var Хато коди : integer);	Бутун ёки ҳақиқий сонни ифодаловчи сатрни сонга айлантириш

Сатр ва белгилар билан ишлаш учун	Изоҳлар
Concat (<i>Сатр 1 : string; ...; Сатр N : string</i>): string	Бир неча сатрни биттага бирлаштириш
Copy (<i>Сатр : string; Белги Рақами : integer</i>): string	Ички сатрни ажратиш
Delete (<i>var Сатр : string; Белги Рақами : integer; Қанча : integer</i>)	Сатр қисмини олиб ташлаш
Length (<i>Сатр : string</i>) : integer	Сатр узунлиги
Pos (<i>Сатр : string; Ички сатр : string</i>) : byte	Ички сатрнинг сатрдаги ўрни
Chr (<i>Белги Коди : byte</i>) : Char	Кўрсатилган коддаги белги

Функция ва процедуралар баёнида қуйидаги белгилашлар қабул қилинган:

- функция ва процедуралар исми ярим йўғон шрифт билан ажратилган;
- параметрлар курсив билан ажратилган; (параметр сифатида ўзгармаслар, ўзгарувчилар ёки тегишли турдаги ифодалар ишлатилиши мумкин. Агар параметр сифатида асосий дастурнинг ўзгарувчиси бўлиши шарт бўлса, унинг олдида, албатта, **var** сўзи, параметрдан кейин унинг тури ёзилиши керак);
- шарт бўлмаган параметрлар квадрат қавсларда ёзилган;

- функция параметрлари рўйхатидан кейин (икки нукта орқали:) функция қайтарадиган натижа тури кўрсатилади.

1.2.3.1. МАТЕМАТИК ФУНКЦИЯЛАР

1. Abs

Синтаксис;

function Abs (x)

Бутун ёки ҳақиқий турдаги аргументнинг абсолют қийматини қайтаради.

2. Arctan

Синтаксис;

function Arctan ($x : \text{real}$) : real;

Радианларда ифодаланган аргумент бурчак миқдорининг арктангенсини ҳисоблайди.

3. Cos

Синтаксис:

function Cos ($x : \text{real}$) : real;

Радианларларда ифодаланган аргумент бурчак миқдорининг косинусини ҳисоблайди.

4. EXP

Синтаксис:

function EXP ($x : \text{real}$) : real;

Аргумент экспонентасига тенг қийматни ҳисоблайди.

5. Ln

Синтаксис:

function Ln ($x : \text{real}$) : real;

Аргументнинг натурал логарифмига тенг бўлган қийматни ҳисоблайди.

6. Sin

Синтаксис:

function Sin ($x : \text{real}$) : real;

Радианларда ифодаланган аргумент бурчак миқдорининг синусини ҳисоблайди.

7. SQR

Синтаксис:

function SQR (x);

Бутун ёки ҳақиқий турдаги ифода аргументининг квадратини ҳисоблайди.

8. SQRT

Синтаксис:

function SQRT (x : real) : real;

Аргументдан квадрат илдиз чиқарилган қийматни аниқлайди.

9. Random

Синтаксис:

function Random [(соҳа : word)]

Агар Соҳа параметри кўрсатилмаган бўлса, $0 \leq x \leq 1$ шартни қаноатлантирувчи X тасодифий сонни беради. Агар Соҳа параметри кўрсатилган бўлса, функция $0 \leq x \leq$ Соҳа шартни қаноатлантирувчи **Word** туридаги тасодифий сонни қабул қилади.

Изоҳ: **Random** функциясига биринчи марта мурожаат қилишда, **Randomize** процедурасини чақириш ёрдамида тасодифий сонлар дастур генераторини инициализация қилиш зарур.

10. Randomize

Синтаксис:

procedure Randomize;

Тасодифий сонлар дастур генераторини инициализациялайди.

1.2.3.2. ЎЗГАРТИРИШНИНГ ФУНКЦИЯ ВА ПРОЦЕДУРАЛАРИ

1. INT

Синтаксис:

function Int (x : real) : real;

Ҳақиқий турдаги қиймат сифатида аргументга бутун қисмни беради. Шакл алмаштиришда аргументнинг каср қисми ҳисобга олинмайди, яъни функция яхлитламайди.

2. Round

Синтаксис:

function Round (x : real) : real;

Аргументни энг яқин бутун сонга яхлитлайди.

3. Str

Синтаксис:

procedure Str (*x* [:2 *Белгилар Жами* [:*Каср Қисм*]]; **var** *Camp* : **string**);

Сонли ифодани унинг сатрли баёнига айлантиради. *Белгилар Жами* ва *Каср Қисм* – бутун турдаги белгиларнинг умумий сонини ва сон тасвиридаги каср қисм белгилари миқдорини берувчи, шарт бўлмаган, ифодадир.

4. Trunc

Синтаксис:

function Trunc (*x* : **real**) : **longint**;

Аргументнинг бутун қисмини бутун турдаги қиймат сифатида беради. Шакл алмаштиришда аргументнинг каср қисми ҳисобга олинмайди, яъни функция яхлитланмайди.

5. Val

Синтаксис:

Procedure Val (*Camp* : **string**; **var** *Ўзгарувчи*; **var** *Хато* : **integer**);

Бутун ёки ҳақиқий сонни ифодаловчи сатрни сонга айлантиради. Ҳосил бўлган қиймат процедурани чақиришда кўрсатилган ўзгарувчи томонидан ўзлаштирилади.

«*Хато*» ўзгарувчи, агар шакл алмаштириш бажарилмаса, унинг амалга ошмаслигига сабаб бўлган сатр белгиси рақамини ўзлаштиради. Агар шакл алмаштириш муваффақият билан бажарилса, «*Хато*» нинг қиймати нолга тенг.

1.2.3.3. САТР ВА БЕЛГИЛАР БИЛАН ИШЛАШ УЧУН ФУНКЦИЯЛАР ВА ПРОЦЕДУРАЛАР

1. Chr

Синтаксис:

function Chr (*Белги Коди* : **byte**) : **char**;

Кўрсатилган код билан белгини қабул қилади.

2. Concat

Синтаксис:

function Concat (*S1* [, *S2*, ..., *SN*] : **string**) : **string**;

Функцияни чақиришда кўрсатилган қаторлар бирлашмаси бўлган сатрни қабул қилади

3. Copy

Синтаксис:

function Copy (*s* : **string**; *n* : integer; *l* : integer) : **string**;

S сатр қисми - ички сатрни қабул қилади. Ички сатр *n* – рақамли белгидан бошланади ва *l* белгидан иборат бўлади.

4. Delete

Синтаксис:

procedure Delete (**var** *S*: **string**; *Белги Рақами*:integer; *Қанча*:integer);

S сатрдан *n* рақами билан бошланувчи ва *l* белгидан иборат бўлган қисмни ўчиради.

5. Length

Синтаксис:

function Length (*camp* : **string**) : integer;

Аргумент – сатр белгилари миқдорига тенг қийматни қайтаради.

6. POS

Синтаксис:

function POS (*camp* : **string**; *Ички camp* : **string**) : byte;

Сатрдаги ички сатр ўрнини (белги рақамини) қайтаради.

7. Сатр ва белгилар ишлатилган дастурларда кўпинча график режимнинг DetectGraph процедураси ишлатилади.

DetectGrap

Синтаксис:

DetectGrap (**var** *Драйвер*, *Режим* : integer);

График адаптерни текширади ва қайси график драйвер ва режим ишлатилаётганини аниқлайди.

1.2.3.4. ФАЙЛЛАР БИЛАН ИШЛАШ УЧУН ПРОЦЕДУРА ВА ФУНКЦИЯЛАР

1. Append

Синтаксис:

procedure Append (**var** *F* : text);

Файл охирига кўшиш режимида файлли ўзгарувчи *F* билан боғлиқ мавжуд файлни очади. Мавжуд бўлмаган файлни очишга уринишда дастурни бажариш вақти хатоси пайдо бўлади.

2. Assing

Синтаксис:

procedure Assing (**var** *F* ; *Файл Исми* : **string**);

F файл ўзгарувчини аниқ файл билан боғлайди.

3. Close

Синтаксис:

procedure Close (**var** *F*);

F файл ўзгарувчи билан боғлиқ бўлган файлни ёпади.

4. EOF

Синтаксис:

function EOF (**var** *F*) : boolean

F файл ўзгарувчи билан боғлиқ файлдан ўқишда файлнинг охирига (**end of file**) етганлигини текширади. Агар ўқув кўрсаткичи файл охирига етган бўлса, функция EOF – TRUE, акс ҳолда – FALSE қийматни олади.

5. Erase

Синтаксис:

procedure Erase (**var** *F*);

Исми F файл ўзгарувчи билан боғлиқ файлни йўқотади.

6. IOResult

Синтаксис:

function IOResult : integer;

Охирги бажарилган файлни киритиш – чиқариш (шу жумладан файлни очиш ва ёпиш) иши натижаси кодини қабул қилади. Агар киритиш-чиқариш иши муваффақиятли бажарилган бўлса, функция нолни қайтаради.

Изоҳ: Дастур IOResult функциясини ишлата олиши учун киритиш-чиқариш натижасида хато пайдо бўлиши мумкин бўлган кўрсатмалар олдида {\$I -}, кўрсатмалардан кейин эса {\$I +} директиваларни жойлаштириш керак.

7. Reset

Синтаксис:

procedure Reset (**var** *F* [: **file**; Recsize : word]);

Мавжуд файлни очади. Файл тури ихтиёрий бўлиши мумкин. Агар файл элементлари стандарт турларнинг бирортасига тегишли бўлмаса, **Recsize** параметр ёзув узунлигини (байтларда) беради. Мавжуд бўлмаган файлни очишга уринишда бажариш вақти хатоси пайдо бўлади.

8. Rewrite

Синтаксис:

procedure Rewrite (var F [: file; Recsize : word]);

Исми F файл ўзгарувчи билан боғлиқ бўлган файлни яратади ва очади. Агар шундай исмли файл бор бўлса, Rewrite процедура уни йўқотади ва янгисини яратади.

1.2.3.5. БОШҚА ФУНКЦИЯЛАР

1. ClrEol

Синтаксис:

procedure ClrEol;

Жорий (курсор турган) сатрни, курсор турган хонадан сатр охиригача TextBackGrand процедураси берган ранг билан бўяб, тозалайди.

2. ClrScr

Синтаксис:

procedure ClrScr;

Экраннинг жорий ойнасини (Windows процедураси ўрнатган) TextBackGrand процедураси берган ранг билан бўяб, тозалайди.

3. Delay

Синтаксис:

procedure Delay (Тўхтатиши : word);

Дастур бажарилишини миллисекундларга тўхтатишни таъминлайди.

4. Dispose

Синтаксис:

procedure Dispose (var P);

Динамик ўзгарувчи эгаллаган P кўрсатаётган хотирани бўшатади.

5. Eoln

Синтаксис:

function Eoln [(var F : text)] : boolean;

Агар Eoln функцияга мурожаат қилинганда, параметр кўрсатилмаган бўлса, функция клавиатуранинг буферидан ўқиладиган навбатдаги белги «янги сатр» белгиси бўлиш – бўлмаслигини текширади. Агар бўлса, функция TRUE, акс ҳолда FALSE қийматларини қабул қилади. Агар параметр кўрсатилган бўлса, функция худди шундай тартибда, функцияни чақиришда кўрсатилган, файл ўзгарувчи билан боғлиқ бўлган матнли файлнинг навбатдаги белгисини текширади.

6. GotoXY

Синтаксис:

procedure GotoXY (*x, y* : byte);

(*x, y*) координатага эга бўлган экраннинг нуқтасига курсорни кўчиради.

7. Halt

Синтаксис:

procedure Halt [(*Тугатиш коди* : word)];

Дастур бажарилишини тугатади ва бошқаришни операцион тизимга узатади.

8. New

Синтаксис:

procedure New (*var P*);

Динамик ўзгарувчи учун хотира ажратади ва *P* кўрсаткич учун ажратилган соҳа адресини беради.

9. ParamCount

Синтаксис:

function ParamCount : word;

Буйруқ сатрига параметрлар миқдорини беради.

10. ParamStr

Синтаксис:

function ParamStr (*N* : word) : **string**;

Функцияга мурожаат қилганда рақами кўрсатилган буйруқ сатри параметрини қабул қилади. Бажарилаётган дастур файлига йўл ва унинг исми (масалан, `c:\tp\exe@tpu\myprog.exe`) ParamStr (0) нинг қиймати бўлади.

11. ReadKey

Синтаксис:

function ReadKey : char;

Босилган клавишга мос белгини беради. Ord функциясидан фойдаланиб, босилган клавиш кодини ҳосил қилиш мумкин. Агар хизмат клавиши босилган бўлса (масалан, курсорни силжитиш клавиши), ReadKey функция 0 кодлик белгини беради. Бу ҳолда хизмат клавишини ўхшатиш учун яна бир марта ReadKey функцияни чақириш керак, бу ҳолда у хизмат коди клавишини беради. Қуйида тез-тез ишлатиладиган баъзи бир клавишларнинг кодлари келтирилган.

Клавиш	Код
<Esc>	27
<Backspace>	8
<Enter>	13
<Space>	32
<↑>	0; 72
<↓>	0; 80
<←>	0; 75
<→>	0; 77

12. TextBackGround

Синтаксис:

procedure TextBackGround (*ранг* : byte);

Write ва writeln кўрсатмалари билан чиқариладиган ахборотлар таг рангини беради. Ранг параметри сифатида қуйида номланган ўзгармасларнинг биронтаси ишлатилиши мумкин.

Ўзгармас	Ранг	Ранг рақами
Black	Қора	0
Blue	Кўк	1
Green	Яшил	2
Cyan	Фируза	3
Red	Қизил	4
Magenta	Бинафша	5
Brown	Жигарранг	6
LightGray	Оқ	7

13. TextColor

Синтаксис:

procedure TextColor (*ранг* : byte);

Write ва writeln кўрсатмалари билан чиқариладиган ахборот белгиларининг рангини ўрнатади. Ранг параметри сифатида қуйида келтирилган номланган ўзгармасларнинг биридан фойдаланиш мумкин:

Ўзгармас	Ранг	Ранг рақами
Black	Қора	0
Blue	Кўк	1
Green	Яшил	2

Cyan	Фируза	3
Red	Қизил	4
Magenta	Бинафша	5
Brown	Жигарранг	6
LightGray	Оқ	7
DarkGray	Кулранг	8
LightBlue	Ҳаворанг	9
LightGreen	Очиқ яшил	10
LightCyan	Очиқ фируза	11
LightRed	Очиқ қизил (алвон ранг)	12
LightMagenta	Очиқ бинафша	13
Yellow	Сарик	14
White	Оқ, яркирок	15

14. Where X

Синтаксис:

function Where X : byte;

Window процедураси билан берилган курсорнинг жорий ойнадаги X координатасини очади.

15. Where Y

Синтаксис:

function Where Y : byte;

Window процедураси билан берилган курсорнинг жорий ойнадаги Y координатасини очади.

16. Window

Синтаксис:

procedure Window (x1, y1, x2, y2 : byte);

Экран соҳаси – ойнани аниқлайди. x1, y1 координаталар ойнанинг юқори чап бурчаги координаталарини, x2, y2 эса ойнанинг ўнг паст бурчаги координаталарини беради.

§ 1.3. ЎЗГАРУВЧИЛАРНИНГ ХОТИРАДА САҚЛАНИШ СОҲАСИ

Компьютер хотирасида сонлар битлар кўринишида сақланади. Ҳар бир бит битта иккилик рақамига тенг қиймат (0 ёки 1) қабул қилади. Саккиз бит бир байтга тенг. Саккизта иккилик рақам ёрдамида ёзиладиган максимал сон - 11111111, бу 255 сонига мос, энг кичик сон эса нол. Шунинг учун байт 0 дан 255 гача қиймат қабул қилиши мумкин.

Ўзгарувчилар хотирада сақланади. Ҳар хил турдаги ўзгарувчилар ҳар хил қийматлар қабул қилиши мумкинлиги учун уларни сақлашга хотиранинг ҳар хил сифими керак. Ўзгарувчи учун хотира байтларга ажратилади (бутун сон). Масалан, *char* туридаги ўзгарувчи қиймат 256 та белгидан ихтиёрий биттаси бўлиши мумкин. Шунинг учун бундай турдаги ўзгарувчини сақлаш учун битта байт етарлик. *Integer* бутун турдаги ўзгарувчининг қиймати бўлиб – 32768 дан 32767 (65535 та қиймат) гача сон бўлиши мумкин. Бу турдаги ўзгарувчини сақлаш учун 2 байт керак. Демак, тур қиймати соҳаси қанча катта бўлса, шу турдаги ўзгарувчини сақлаш учун шунча кўп байт керак:

Ўзгарувчи тур	Эгаллайдиган хотира, байт	Қийматлар соҳаси
<i>CHAR</i>	1	Ихтиёрий белги
<i>STRING</i>	256	256 тагача белгига эга сатр
<i>STRING [N]</i>	1 x n	n тагача белгига эга сатр
<i>BYTE</i>	1	0-255
<i>WORD</i>	2	0-65535
<i>INTEGER</i>	2	-32768-32767
<i>LONGINT</i>	4	-2147483648-2147483647
<i>REAL</i>	6	2.9E – 39 – 1.7e 38
<i>SINGLE</i>	4	1.5E – 45 – 3.4e 38
<i>DOUBLE</i>	8	5.0E – 324 – 1.7e 308
<i>EXTENDED</i>	8	3.4E – 4932 – 1.1e 4932

Дастурда бир хил қиймат учун ҳар хил турдаги ўзгарувчилардан фойдаланиш мумкин, бунда улар хотиранинг ҳар хил ҳажмини эгаллайди. Масалан, агар дастурда *Day* (ой куни) ўзгарувчиси ишлатилаётган бўлса, унинг учун *byte*, *integer* ёки *longint* турларини бериш мумкин. Биринчи ҳолда хотиранинг бир, иккинчисида – икки, учинчисида – тўрт байти банд бўлади. Лекин фақат битта байтдан тўлиқ фойдаланилади, қолганлари эса фақат банд қилинади. Шунинг учун ҳар бир ўзгарувчи учун энг тўғри келадиган турни танлаш керак. Сатр ўзгарувчилари ва массивларни баён этишга алоҳида эътибор бериш керак. Сатр ўзгарувчиларига жой ажратилаётганда, агар сатрнинг чегаравий узунлиги берилмаган бўлса, ўзгарувчига 256 байт ажратилади. Масалан, ўзгарувчини эълон қилишда **name : string** деб эмас, балки **name : string [30]** деб ёзиш керак.

Дастурнинг ҳар бир массиви учун жой ажратилади, жой ҳажми массив элементлари тури ва уларнинг миқдори билан аниқланади. Масалан, икки ўлчамли (масалан, 20x20) массивнинг ҳақиқий сонларини сақлаш учун хотиранинг 3 килобайтдан кўпроқ сизими керак ($20 \times 20 \times 8 = 3200$).

Компьютер хотираси ҳажми чегарасиз бўлиб кўрингани билан, у самарасиз ишлатилса, етишмай қолиши мумкин.

§ 1.4. ЭҲМДА ПАСКАЛ ТИЛИДА ЁЗИЛГАН ДАСТУРЛАР БИЛАН ИШЛАШ ТАРТИБИ

Бош менюга кириш:

F10 клавиши босилади.

Менюдан еки мулоқот ойнасидан, ҳеч бир иш бажармасдан, чиқиш:

ESC клавиши босилади.

Бош менюнинг ихтиерий ички менюсини тез очиш: **Alt** клавиши босилади ва ушлаб турилади, у кўйиб юборилмасдан меню номланишининг биринчи ҳарфи босилади. Масалан, **Alt + F** - экранда **File** менюсини очади.

Меню буйруқларини тез чиқариш. Очилган меню буйруқларини зудлик билан чиқариш учун буйруқ номидаги ранги билан ажралиб турган ҳарф езилган клавишни босиш керак. Масалан, очилган **File** менюсида "A" клавиши босилса, **Save AS...** буйруғи чиқарилади. Еки бу тезкор клавишлар ёрдамида ҳам бажарилиши мумкин:

Alt + F A

Ишчи (жорий) каталогни ўзгартириш.

1. **File** менюси очилади.
2. **Change dir...** буйруғи чиқарилади. Диалог ойна очилади, унинг Directory name майдонида жорий каталог номи, Directory tree майдонида эса жорий дискнинг каталоглар шажараси кўрсатилган бўлади.
3. Directory name майдонига янги жорий каталог номи езилади, **ТАВ** клавиши езилган каталоглар шажарасида унинг исми кўрсатилади.

4. Танланган каталог **Enter** клавишини босиш билан мустаҳкамланади.
5. **TAB** клавишини босиш билан курсор ОК тугмага ўрнатилади ва **Enter** босилади

Янги дастурни киритиш учун ойна очиш.

1. **File** менюси очилади.
2. **New** буйруғи чақирилади.

Дискка янги дастурни езиш.

1. **File** менюси очилади.
2. **Save as...** буйруғи чиқарилади. **Save FileAs** мулоқот ойнаси очилади, бу ерда **Save FileAs** майдони янги дастур учун файл номи бириктирилади. Шунингдек бу ойнада янги дастурни мавжуд файллардан бирининг номи билан аташ учун ҳам имконият бўлади. Бунинг учун **TAB** клавишини босиш (курсор File майдонига ўтади) ва мавжуд рўйхатдан файл исмини танлаш керак.
3. **Enter** клавиши босилади. Мавжуд исм билан ёзилаётганда тегишли огоҳлантириш берилган ва файлни қайта ёзиш ёки уни танланган исм билан езишдан воз кечиш имконияти бўлади.

Файлни тахрир қилиш учун очиш.

1. **File** менюси очилади.
2. **Open...** буйруғи чиқарилади (**F3** клавиш ҳам бу ишни бажаради).
3. Экранда **Open a File** ойнаси пайдо бўлади. Name майдони Files майдони чиқариладиган файлларни танлаш учун шаблонга эга бўлади. сукут ҳолатида *.pas шаблони тавсия этилади. Тегишли шаблондаги файллар рўйхати Files майдони, дискнинг жорий каталогида мавжуд бўлган файллардан танлаб чиқарилади.
4. Бу босқичда куйидаги ишлар бажарилади:
 - 4а. **TAB** клавиши босилади, курсор File майдонига ўтади, мавжуд рўйхатдан файл номи танланади.
 - 4б. Name майдонидаги шаблон ўзгартирилади, шундай кейин 4а босқичдаги иш бажарилади.
 - 4в. Очиладиган файлнинг номи бевосита Name майдонига киритилади.
5. **Enter** клавиши босилади. 4а ва 4б ҳолларда клавиш икки марта босилади.

Фаол ойнадаги дастурни эски ном билан сақлаш.

I - усул. **F2** клавиши босилади.

II - усул. 1. **File** менюси очилади.

2. **Save** буйруғи чақирилади.

Фаол ойнадаги дастурни бажаришга юбориш.

Ctrl + F9 клавишлари босилади.

Натижани чиқариш учун экранни тозалаш.

Экранни тозалаш учун дастурда қуйидагиларни бажариш зарур:

- uses таклифида Crt стандарт модули киритилади (Uses Crt):
- дастурнинг оператор блоки бошида ClrScr процедураси чақирилади.

Мисол:

```
Program PR;  
Uses Crt;  
var A: integer;  
begin  
    ClrScr;  
    Readln (A);  
    Writeln ('A = ', A);  
end.
```

Янги дастурни яратиш ва хатоларини тузатишда бажариладиган ишларининг намунавий режаси.

1. Янги дастур киритиш учун таҳрирлашнинг янги ойнаси очилади.
2. Янги дастур матни терилади.
3. Дастур матни дискка езилади (янги дастур матни ишга туширилгунига қадар дискка езилиши зарур).
4. Дастур бажаришга юборилади (Ctrl + F9).
5. Агар дастурда синтаксис хатоларга йўл қўйилган бўлса, экранда тегишли ахборот пайдо бўлади, курсор эса хато жойни кўрсатади. Бундай пайтда таҳрир буйруқлари ёрдамида тузатилиши ва яна дастур бажаришга юборилиши керак.
6. Дастур натижаларини кўриш (Alt + F5).
7. Агар хато натижалар олинган бўлса, йўл қўйилган алгоритмик хатоларни тузатиш ва дастурни яна бажаришга юбориш керак.

8. 4 - 7 - босқичлар тўғри ечимлар олингунча такрорланади.
9. Тузатилган дастур дискда сақланади (F2).

Паскал тилидаги дастур билан ишлашда тезкор клавишлар.

Умумий қўлланишдаги клавишлар.

F10 - бош менюга кириш.

ESC - диалог еки меню ойнасини епиш.

Alt + X - Паскал дастури билан ишлаш ойнасидан чиқиш.

Ctrl + Break - ишга туширилган дастурни тўхтатиб, бошланғич ойнага қайтиш (ишга тушган дастур қотиб қолганда еки епиқ циклга тушиб қолинганда бу клавишлар босилади).

Print Screen - экран нусхасини принтерга чоп этиш.

Pause - ихтиерий клавиш босилгунча ўзгараётган тасвирни экранда тўхтатиб туриш.

Ёрдам тизими билан ишлаш клавишлари.

F1 - ёрдам тизимининг шу дақиқада фаол бўлган ойна ёки курсор кўрсатаётган меню буйруғи тўғрисидаги ахборотини ёритади.

F1 - (икки марта) - ёрдам тизимидан фойдаланиш тўғрисидаги кўрсатмани экранга чиқаради.

Ctrl + F1 - фаол ойнадаги курсор турган ибора ҳақидаги ахборотни экранга чиқаради.

Alt + F1 - Help ойнасининг олдинги ойнасига қайтиш. Бу буйруқни кўп марта бажарилиши тесқари тартибда Helpнинг 20 тагача охириги ойналарини чиқаради.

Shift+F1 - ёрдам тизимида мавжуд бўлган ибораларнинг алфавит тартибидаги рўйхатига эга бўлган Index ойнасини экранга чиқаради.

Файлларни очиш, сақлаш ва таҳрир ойналари билан ишлаш клавишлари.

F2 - таҳрирлашнинг фаол ойнасидаги дастурни дискдаги файлга эски ном билан сақлаш.

F3 - таҳрирлаш ва ишга тушириш учун зарур бўлган файлни дискдан танлаш учун Open a File мулоқот ойнасини чиқариш.

Alt + F3 - таҳрирлашнинг фаол ойнасини епиш.

F6 - бир неча марта **F6** клавишнинг кетма - кет босилиши очилган ойналар фаоллиги алмашинувининг такрорланишини келтириб чиқаради.

Shift + F6 - F6 клавишига ўхшаб ишлайди, лекин ойналарни тескари кетма - кетликда алмаштириб боради.

Alt + O - Паскал ишга туширилган лаҳзадан очилган ҳамма ойналар кўрсатилган Window List мулоқот ойнасини очади.

F5 - фаол ойнани тўла экран ўлчамида очади, агар ойна шу ўлчамда бўлса, унинг бошланғич ҳолатини тиклайди.

Ctrl + F5 - фаол ойна ўлчамини ва (ёки) унинг экрандаги ўрнини ўзгартириш. Ойнанинг экран бўйича силжиши йўналиш клавишлари ёрдамида, ўлчамни ўзгартириш эса **Shift** ва йўналиш клавишларининг бир вақтда босилиши билан бажарилади

Дастур матни бўлаги билан ишлаш клавишлари.

Shift + "йўналиш клавишлари" - дастур бўлагини ажратади.

Shift + Del - дастурнинг ажратилган бўлагини ўчиради ва Clipboard буферига жойлаштиради.

Ctrl + Ins - дастурнинг ажратилган бўлагини Clipboard буферига нусхалайди.

Shift + Ins - Clipboard даги ажратилган бўлакни ойнадаги курсор жойлашган хонага қўяди.

Ctrl+Del - ажратилган бўлакни Clipboard буферига узатмасдан ўчиради.

Alt + BackSpace - таҳрирлашдаги охириги ишни бекор қилади.

Компиляция ва бажаришга юбориш клавишлари.

Alt + F9 - таҳрирлашнинг фаол ойнасидаги файлни компиляция қилиш.

F9 – .exe файлини яратиш билан кўп модулли дастурни шартли компиляция қилиш. Агар охириги компиляция вақтидан баъзи бир модулларга ўзгартиришлар киритилган бўлса, фақат ўзгартирилган ва уларга боғлиқ бўлган модулларгина қайта компиляция қилинади. Оддий дастурлар учун шунга тенг кучли **Alt + F9** босилади.

Ctrl+F9 - таҳрирлашнинг актив ойнасидаги дастурни бажаришга юбориш.

Дастурларни тузатиш клавишлари.

Alt + F5 - дастур бажарилиши натижаларини кўриш.

F8 - дастурнинг қадамба-қадам бажарилиши. Процедура ва функцияларни чақириш битта оператор (битта қадам) каби бажарилади.

F7 - дастурни кадамба-қадам бажариш. Процедура еки функцияларни чақиришда унинг матнига кириш ва операторларни кадамба - қадам бажариш содир бўлади.

F4 - дастур кадамба-қадам бажарилишининг жорий сатридан курсор жойлашган сатргача дастур қисмининг бажарилиши.

Ctrl + F2 - дастурни таҳрир қилиш ишини якунлайди ва уни хотирадан бўшатади.

Ctrl + F3 - Call Stack ойнасини очади. Бу ойнада шу дақиқада бажарилаётган процедурагача чиқарилган дастур процедуралари рўйхати кетма-кетлиги кўрсатилади.

Ctrl + F4 - Evaluate and modify ойнасини очади. Бу ойнада қиймати аниқланиши талаб этилган ифодани кўрсатиш, дастур ўзгарувчилари қийматларини ва берилганлар элементларини қараб чиқиш ҳамда уларни ўзгартириш керак.

Ctrl + F7 - Add Watch мулоқот ойнасини очади. Бу ойнага дастурчи тузатишни бажариш вақтида, қийматлари қизиқтирадиган ифода ёки ўзгарувчи номини кўрсатиши мумкин.

II ҚИСМ. МАСАЛАЛАР ШАРТЛАРИ

§ 2.1. ҚУЙИ МУРАККАБЛИКДАГИ МАСАЛАЛАР

2.1.1. ЎЗГАРУВЧИЛАРНИ ЭЪЛОН ҚИЛИШ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эсда тутиш керак:

- Дастурнинг ҳар бир ўзгарувчиси эълон қилиниши керак;
- Ўзгарувчилар **var** сўзи билан бошланадиган бўлимда эълон қилинади;
- Ўзгарувчиларни эълон қилиш кўрсатмаси қуйидаги кўринишга эга:

Ўзгарувчи Исми: Тури;

- Ўзгарувчи номи сифатида лотин алифбосининг ҳарфлари ва рақам (биринчи белги ҳарф бўлиши керак) ишлатилади;
- **Turbo Pascal** тилининг асосий сонли турлари қуйидагилар: **integer** (бутун), **real** (ҳақиқий каср);
- Ўзгарувчи эълон қилинган кўрсатмадан кейин ўзгарувчининг қўлланилишини кўрсатиш тавсия қилинади.

МАСАЛАЛАР

1. $y=x^2$ функциянинг қийматини ҳисоблаш учун зарур ўзгарувчиларни эълон қилинг.
2. Фунт ўлчовидаги оғирликни килограммга айлантириш учун зарурий ўзгарувчиларни эълон қилинг (бир фунт = 409,5 г).
3. Параллелепипед ҳажмини ҳисоблаш учун зарурий ўзгарувчиларни эълон қилинг.
4. Доира юзасини ҳисоблаш учун зарурий ўзгарувчиларни эълон қилинг ($S=\pi\cdot r^2$).
5. Цилиндр сирт юзаси ва ҳажмини ҳисоблаш учун зарурий ўзгарувчиларни эълон қилинг ($v=2\cdot\pi r(h+r)$; $v=\pi R^2\cdot h$).
6. Бир неча дафтарлар, қаламлар ва чизғичдан иборат хариднинг қийматини ҳисоблаш учун зарурий ўзгарувчиларни эълон қилинг.

2.1.2. ЎЗЛАШТИРИШ КЎРСАТМАСИ*

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- Ўзлаштириш кўрсатмаси ўзгарувчилар қийматини ўзгартириш, шунингдек, формула бўйича ҳисоблашда ишлатилади;
- Ўзлаштириш кўрсатмасининг ўнг томонидаги ифода тури чап томондаги ўзгарувчи турига мос келиши керак (акс ҳолда компилятор – **Type Miss match** – турлар мос келмайди – деган ахборот чиқаради).

МАСАЛАЛАР

7. Ўзгарувчига 1,5 қийматни берувчи кўрсатмани ёзинг.
8. **Summa** ўзгарувчисига нол қиймат берувчи кўрсатмани ёзинг.
9. n ўзгарувчи қийматини 1 га оширувчи кўрсатмани ёзинг.
10. **counter** ўзгарувчи қийматини 2 га камайтирувчи кўрсатмани ёзинг.
11. X_1 ва X_2 ўзгарувчилар қийматининг ўрта арифметигини топиш кўрсатмасини ёзинг.
12. $y = -2,7x^3 + 0,23x^2 - 14$ функция қийматини ҳисоблаш кўрсатмасини ёзинг.
13. Масофани километрдан верстларга ўтказиш формуласи бўйича ҳисоблаш кўрсатмасини ёзинг (бир верст = 1066,8 м).
14. Тўртбурчак юзини ҳисоблаш кўрсатмасини ёзинг.
15. Учбурчак юзини ҳисоблаш кўрсатмасини ёзинг.
16. Шар сирт юзаси ва ҳажмини ҳисоблаш кўрсатмасини ёзинг
($v = \frac{3}{4}\pi R^3$; $S = 4\pi R^2$).
17. Ғовак цилиндр ҳажмини ҳисоблаш кўрсатмасини ёзинг
($v = \pi \cdot R / R_1^2 - R_2^2$).
18. Конус ҳажмини ҳисоблаш кўрсатмасини ёзинг. ($S = \frac{1}{3} \cdot \pi \cdot R^2 \cdot h$).

2.1.3. ЧИҚАРИШ КЎРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эсланг:

* Баъзи бир адабиётларда кўрсатма ибораси ўрнида оператор сўзи ишлатилади.

- **write** ва **writeln** кўрсатмалари монитор экранига ахборотлар ва ўзгарувчи қийматларини чиқариш учун ишлатилади;
- битта **write** (**writeln**) кўрсатмаси экранга бир неча ўзгарувчи қийматларини ва (ёки) бир неча ахборотларни чиқариши мумкин;
- параметрсиз **writeln** кўрсатмаси курсорни экраннинг навбатдаги сатр бошига ўтказди.

МАСАЛАЛАР

19. Эcranга исм ва фамилияни чиқарувчи дастурни ёзинг.

20. Эcranга қуйидаги тўртликни чиқарувчи дастурни ёзинг.

«Инсон ақли - галактика йўлида чирок,
Балки космос мозорида кўмилар таним?
Йўқ – йўқ, инсон даричасин очган у чокда,
Тарихларга-у, ўлмас, деб ёзди, Ватаним».

Т.Ҳамид

21. Каср қисмида учта рақам бўлган ўзгарувчи қийматини экранга чиқариш кўрсатмасини ёзинг.

22. Қуйидаги кўрсатмалар нима қилишини аниқланг:

write (a);

write ('a');

23. *a*, *b* ва *c* каср ўзгарувчи қийматларини бир сатрга чиқарувчи кўрсатмани ёзинг. Ҳар бир сон ўнлик нуқтадан кейин икки хона аниқлик билан чиқарилиши керак (**writeln** (a:6:2, `...`, в:6:2, `...`, с:6:2)).

24. *a*, *b* ва *c* каср ўзгарувчи қийматларининг ҳар бирини битта сатрдан ўзгарувчи номлари кўрсатилган кўринишда чиқаринг.

ҚЎШИМЧА КЎРСАТМАЛАР

Crt стандарт кутубхона **write** ва **writeln** кўрсатмалари билан чиқариладиган ўзгарувчиларга таг ранг ва белги рангларини бера олувчи процедураларга эга. Бу бўлим масалаларини ечишга ўтишдан олдин, қуйидагиларни эшлаш керак:

- **Crt** кутубхона процедура ва функцияларини ишлата олиш учун дастурдаги биринчи кўрсатма **Uses Crt** бўлиши керак;
- белгилар рангини **TextBackGround** процедураси ўрнатади;
- таг рангини **TextColor** процедура белгилайди;

- **TextBackGround** ва **TextColor** процедураларини ишлатишда ранг танланган ўзгарувчи ёки бутун ўзгармаслар билан берилиши мумкин;
- **Clr Scr** процедура экранни тозалайди;
- Дастур матнига, **ClrScr**, **TextBackGround** ва **TextColor** процедураларидан фойдаланиш мумкин бўлиши учун **Uses Crt** директивасини киритиш керак.

МАСАЛАЛАР

25. Қуйидаги тўртликни кулранг ҳарфлар билан кўк рангли юзага чиқариш дастурини ёзинг.

«Дунё бамисоли бир заррин кўза,
Суви гоҳ аччиғу, гоҳ ширин бўза.
Эй, инсон, умрингга бино қуйма кўп,
Қазойи – муаллақ турар бош узра!»

Паҳлавон Маҳмуд

26. Қуйидаги камалак рангини билдирувчи сўзлар кетма-кетлигини экранга чиқариш дастурини ёзинг: кизил, олов ранг, сариқ, яшил, ҳаворанг, кўк, бинафша ранг. Ҳар бир ранг номини билдирувчи сўз шу рангли ҳарфлар билан ёзилиши керак.

2.1.4. КИРИТИШ КЎРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- Дастур билан ишлаш вақтида бошланғич қийматларни клавиатурадан киритиш учун **readln** кўрсатмаси берилган;
- Битта **readln** кўрсатмаси билан бир нечта ўзгарувчиларни киритиш мумкин;
- Дастур иши вақтида киритилмаган маълумотлар тури **readln** кўрсатмаси айtilган ўзгарувчи турига мос келиши керак;
- Ўзгарувчининг клавиатурадан киритилаётган қиймати киритилган маълумотлар турига мос келмаса, дастур ишини тугатади ва экранга **Error 106 : Invalid numeric format** (агар, дастур **Turbo Pascal** муҳитидан ишга туширилган бўлса) ёки **Run time error 106** (агар дастур операцион системадан ишга туширилган бўлса) деган ахборот чиқади.

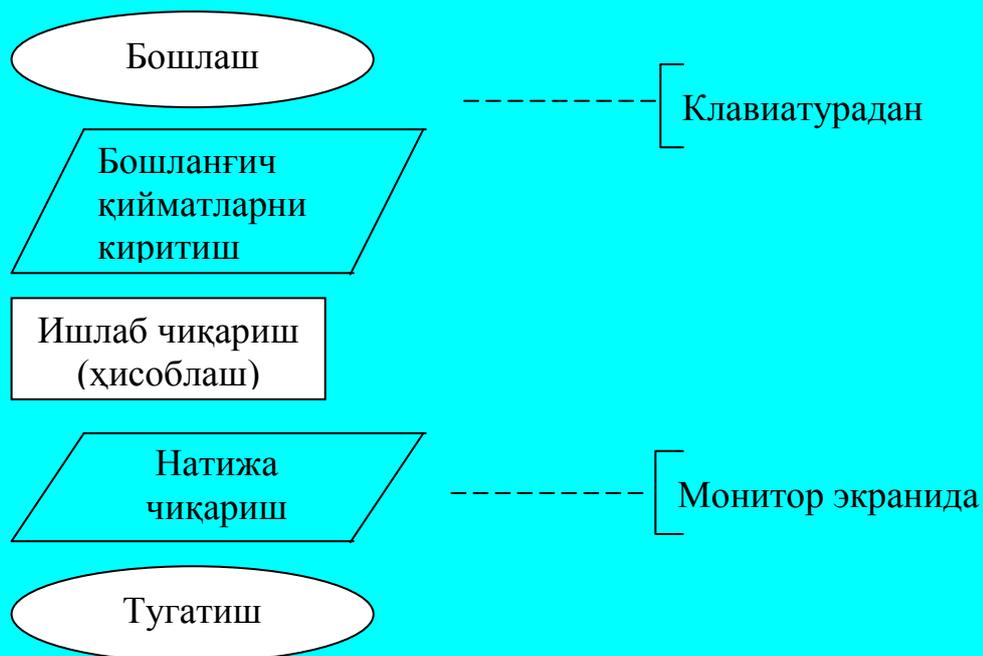
МАСАЛАЛАР

27. U ва r ўзгарувчилар қийматини киритишни таъминловчи кўрсатма ёзинг. Дастур ишлаш вақтида дастурчи ҳар бир сонни терганидан кейин <Enter> клавишини босади, деб ҳисобланади. Фойдаланувчи сонларни ҳар хил сатрдан киритади.
28. U ва r ўзгарувчилар қийматини киритишни таъминловчи кўрсатма ёзинг. Фойдаланувчи сонларни бир сатрдан киритади.
29. Зарур ўзгарувчиларни эълон қилинг ва бошланғич маълумотларнинг киритилишини, цилиндр ҳажмини ҳисоблашни таъминлайдиган дастур бўлагини ёзинг.

2.1.5. ЧИЗИҚЛИ ТАСНИФЛИ ДАСТУРЛАР

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- Чизиқли таснифли дастурлар энг оддий дастурлардир ва улар, одатда, формула бўйича ҳисоблашларни амалга оширишда ишлатилади;
- Чизиқли таснифли дастурларда кўрсатмалар бир-биридан кейин чизиқли кетма-кетликда бажариб борилади;
- Чизиқли таснифли дастур алгоритми қуйидагича баён этилиши мумкин:



Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

30. Параллелограмм юзасини ҳисоблаш дастурини тузинг. Эcranга натижаларнинг қуйидагича чиқарилиши тавсия этилади:

Параллелограмм юзасини ҳисоблаш.

Бошланғич қийматларни киритинг:

Узунлик (см) → **9**

Кенглик (см) → **7.5**

Параллелограмм юзи: 67.50 см^2 .

31. Параллелепипед сирт юзасини ҳисоблаш дастурини тузинг. Қуйида дастур ишлаш вақтидаги экраннинг тавсия қилинадиган кўриниши берилган Параллелепипед сирт юзасини ҳисоблаш.

Бошланғич қийматларни киритинг:

Узунлик (см) → **9**

Кенглик (см) → **7.5**

Баландлиги (см) → **5**

Сирт юзаси: 90.00 см^2 .

32. Цилиндр ҳажмини ҳисоблаш дастурини ёзинг. Қуйида дастур ишлаш вақтидаги экраннинг тавсия қилинадиган кўриниши берилган.

Цилиндр ҳажмини ҳисоблаш.

Бошланғич қийматларни киритинг:

Асос радиуси (см) → **5**

Цилиндр баландлиги (см) → **10**

Дастур ишини тугатиш учун **<Enter>**ни босинг.

Цилиндр ҳажми: 1570.80 см^3 .

33. Бир неча дафтар ва қаламлардан иборат харид қийматини ҳисоблаш дастурини ёзинг. Қуйида дастур ишлаш вақтидаги тавсия қилинадиган экран кўриниши берилган.

Харид қийматини ҳисоблаш.

Бошланғич қийматларни киритинг:

Дафтар баҳоси (сўм) → **40**

Дафтарлар сони → **5**

Қалам баҳоси (сўм) → **30**

Қалам сони → **2**

Харид қиймати: 260 сўм .

- 34.**Агар берилган учбурчакнинг икки томони ва улар ўртасидаги бурчаги маълум бўлса, учбурчак юзини ҳисоблаш дастурини ёзинг. Қуйида дастур ишлаш вақтидаги тавсия қилинадиган экран кўриниши берилган.
- Учбурчак юзини ҳисоблаш.
- Учбурчак 2 томони узунлиги: (см) → **25 17**
- Учбурчак томонлари орасидаги бурчак катталиги (градус) → **30**
- Учбурчак юзи: 106.25 см.кв.
- 35.**Иккита параллел уланган қаршиликлардан иборат электр занжир қаршилигини ҳисоблаш дастурини ёзинг. Қуйида дастур ишлаш вақтидаги тавсия қилинадиган экран кўриниши берилган.
- Элементлари параллел уланган электр занжир қаршилигини ҳисоблаш.
- Бошланғич қийматларни киритинг:
- Биринчи қаршилик катталиги (Ом) → **15**
- Иккинчи қаршилик катталиги (Ом) → **20**
- Занжир қаршилиги: 8.57 Ом
- 36.**Автомобилда дала ҳовлига бориш ва қайтиш қийматини ҳисоблаш дастурини ёзинг. Қуйидагилар бошланғич қийматлар: Дала ҳовлигача масофа (кмларда); 100 км масофага кетадиган бензин миқдори, 1 литр бензин баҳоси. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.
- Дала ҳовлига бориш ва қайтиш сафари қиймати.
- Дала ҳовлигача масофа (км) → **67**
- Бензин сарфи (100 км масофага, литр) → **8.5**
- Литр бензин баҳоси (сўм) → **150**
- Дала ҳовлига бориш ва қайтиш қиймати: 1708.5 сўм
- 37.**Цилиндр сирт юзасини ҳисоблаш дастурини ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.
- Цилиндр сирт юзасини ҳисоблаш.
- Бошланғич қийматлар:
- Асос радиуси (см) → **5.5**
- Цилиндр баландлиги (см) → **7**
- Цилиндр сирт юзаси: 431.97 см².
- 38.**Ихтиёрий берилган қийматларда параллелепипед ҳажмини ҳисоблаш дастурини ёзинг.

39.Ихтиёрий берилган қийматларда куб ҳажмини ҳисоблаш дастурини ёзинг.

40.Ихтиёрий берилган қийматларда учбурчак юзини ҳисоблаш дастурини ёзинг.

41.Верстларда берилган масофа узунлигини кмларга айлантириш дастурини тузинг (1 верст = 1066.8 м). Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Масофани верстлардан кмга айлантириш.

Верстлардаги масофани киритинг ва **<Enter>** ни босинг → **100**

100 верст → бу 106.68 км.

42.Картада кўрсатилган иккита аҳоли пункти ўртасидаги масофани ҳисоблаш дастурини ёзинг (карта масштаби 120 км бир смда. Картадаги икки нуқта орасидаги масофа 3,5 см).

43.Омонат маблағлар бўйича даромад миқдорини ҳисоблаш дастурини тузинг. Даромад фоиз (йиллик фоизга нисбатан) ва омонатни сақлаш вақти (кунларда), омонат миқдори дастур ишлаш вақтида берилади. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Омонат бўйича даромадни ҳисоблаш.

Бошланғич қийматларни киритинг:

Омонат миқдори (сўм) → **2500**

Омонат муддати (кунларда) → **30**

Қўшимча фоиз (йиллик) → **20**

Даромад: **41.10** сўм

Омонат муддати охиридаги йиғинди: **2541.10** сўм

44.Минутларда берилган вақт интервали миқдорини соат ва минутларга айлантириш дастурини ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Вақт интервали миқдорини киритинг (минутларда) ва **<Enter>** ни босинг → **150**

150 минут → бу 2соат 30 минут.

45.Клавиатурадан киритилган қаср сонни пул форматига ўтказиш дастурини ёзинг. Масалан 125 сони 12 сўм 50 тийин кўринишига келтирилиши керак.

Сонни пул форматига ўзгартириш.

Қаср сонни киритинг → 23.6

2.1.6. ТАНЛОВ (If)КўРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- **if** кўрсатма дастурнинг кейинги йўлида (бажарилиши керак бўлган кўрсатмалар кетма-кетлигида) иккита йўналишдан бирини танлаш учун ишлатилади;
- кўрсатмалар кетма-кетлигини танлаш шартнинг бажарилишига боғлиқ равишда дастурнинг бажарилиш вақтида амалга оширилади;
- шарт – мантиқий турдаги ифода бўлиб, у иккита қийматдан бирини танлаши мумкин: **True** (ҳақиқат – шарт бажарилади) ёки **False** (ёлғон – шарт бажарилмайди);
- оддий мантиқий **and** (мантиқий «ВА») ва **or** (мантиқий «ЁКИ») амаллари ёрдамида оддий шартлардан мураккабларини тузиш мумкин.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

46. Иккита сонни бўлишдан ҳосил бўладиган бўлинмани ҳисоблаш дастурини тузинг. Дастур фойдаланувчи киритган қийматларнинг тўғрилигини текшириши керак, агар улар нотўғри бўлса (бўлувчи нолга тенг), хато тўғрисидаги ахборот чиқарилсин. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Бўлинмани ҳисоблаш.

Битта сатрдан бўлинувчи ва бўлувчини киритинг,

Кейин **<Enter>** ни босинг → **12 0**

Адашдингиз. Бўлувчи нолга тенг бўлмаслиги керак.

47. Ичи тешик ҳалқа юзасини ҳисоблаш дастурини тузинг. Дастур бошланғич қийматларнинг тўғрилигини текшириши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Ҳалқа юзини ҳисоблаш.

Бошланғич қийматларни киритинг:

Ҳалқа радиуси (см) → **3.5**

Тешик радиуси (см) → 7

Хато! Тешик радиуси ҳалқа радиусидан катта бўлмаслиги керак!

48. Иккита қаршилиқдан иборат электр занжир қаршилигини ҳисоблаш дастурини тузинг. Қаршилиқлар кетма-кет ёки параллел уланган бўлиши мумкин. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Электр занжир қаршилигини ҳисоблаш.

Бошланғич қийматларни киритинг:

Биринчи қаршилиқ миқдори (Ом) → 15

Иккинчи қаршилиқ миқдори (Ом) → 27.3

Уланиш тури (1-кетма-кет, 2-параллел) → 2

Занжир қаршилиги: 9.68 Ом

49. Квадрат тенгламани ечиш дастурини ёзинг. Дастур бошланғич қийматларнинг тўғрилигини текшириши керак. Иккинчи даражали номаълум сон олдидаги коэффициент нолга тенг бўлса, тегишли ахборот чиқиши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Квадрат тенгламани ечиш.

Битта сатрдан коэффициентлар қийматини киритинг ва <Enter>ни босинг → 12 27 - 10

Тенглама илдизлари:

$$x_1 = -25.551$$

$$x_2 = -28.449$$

50. Харидорга имтиёзни ҳисобга олган ҳолда харид қийматини ҳисоблаш дастурини тузинг. Агар харид йиғиндиси 5000 сўмдан катта бўлса 3 %, 10000 сўмдан катта бўлса 5 % дан харидорга ён босилади. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Имтиёзни ҳисобга олган ҳолда харид қийматини ҳисоблаш.

Харид қийматини киритинг ва <Enter>ни босинг → 6400

Сизга 3 % имтиёз берилади.

Имтиёз ҳисобга олинган ҳолда харид жами: 6208.0 сўм

51. Харид жами 1000 сўмдан кўп бўлса, харид қийматига 10 % ён босилади. Шунини ҳисобга олувчи харид қийматини ҳисоблаш дастурини ёзинг.

52. Буюк ўзбек классик шоири А.Навоий туғилган сана тўғрилигини текшириш дастурини тузинг. Хато жавоб берилганда, дастур тўғри

жавобни чиқарсин. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

А.Навоий қачон туғилган?

Санани киритинг ва <Enter>ни босинг → **1563**

Адашдингиз, А.Навоий 1441 йилда туғилган.

53. Фанлар тарихини билиш бўйича билимни текшириш дастурини тузинг. Дастур саволни ва унга учта жавобни чиқариши керак. фойдаланувчи тўғри жавобни танлаши ва унинг рақамини киритиши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Кибернетика фани асосчиси:

1. Андре Мари Ампер.
2. Исаак Ньютон.
3. Норберт Виннер.

Тўғри жавоб рақамини киритинг ва <Enter> ни босинг → **2**

Адашдингиз.

Кибернетика фани асосчиси – Норберт Виннер →.

54.Клавиатурадан киритилган иккита бутун сонни таққослаш дастурини тузинг. Дастур қайси сон катталигини кўрсатиши керак; агар сонлар тенг бўлса, тегишли ахборотни чиқариши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Битта сатрдан иккита бутун сонни киритинг ва <Enter> ни босинг →

34 67

34 сони 67 дан кичик.

55. Кўпайтирилиши керак бўлган иккита бир хил хонали сонларни чиқарувчи, фойдаланувчи жавобини сўровчи, уни текширувчи ва «Тўғри!» еки «Адашдингиз» деган ахборотларни ва тўғри жавобни чиқарувчи дастур тузинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

6 x 7 нечага тенг?

Жавобни киритинг ва <Enter> ни босинг → **56**

Адашдингиз, $6 \times 7 = 42$

56. Иккинчи жаҳон уруши бошланиши санасини билишни текширувчи дастур тузинг. Дастур тўғри жавобни баҳолаши, хато жавобда дастур тўғри жавобни чиқариши керак.

57.Бирор соха бўйича билимни текширувчи дастур тузинг. Дастур саволни ва унга уч хил жавоб вариантыни чиқариши керак. Фойдаланувчи тўғри жавоб рақамини киритиши керак. Тўғри ва хато жавоблар дастур томонидан изоҳланиши керак.

58.100 гача бўлган сонлар билан айириш амалини бажаришга мисол чиқарувчи, фойдаланувчи жавобини текширувчи, унга изоҳлар берувчи дастур тузинг. Қуйида тавсия этиладиган экран кўриниши берилади:

83 – 17 неча бўлади?

Жавобни киритинг ва <Enter> ни босинг.

→ 67

Адашдингиз! $83 - 17 = 66$

59.Киритилган сонинг жуфтлигини текширувчи дастур ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Бутунни сони киритинг ва <Enter>ни босинг → **23**

23 сони - тоқ.

60.Клавиатурадан киритилган сон учга бўлинишини текширувчи дастур тузинг. Жавоблар изоҳли бўлсин. Масалан:

Бутун сонни киритинг ва <Enter> ни босинг.

→ 451

451 сони учга бўлинмайди.

61.Шанба ва якшанба кунги телефон орқали сўзлашувларнинг 20 % арзонлигини ҳисобга олган ҳолда сўзлашувлар қийматини аниқловчи дастур тузинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Телефон орқали сўзлашувлар қиймати:

Бошланғич қийматларни киритинг:

Сўзлашув вақти (бутун сон. Минут) → **3**

Ҳафта куни (1-душанба, ..., 7-якшанба) → **6**

20 % арзонлаштирилади.

Сўзлашувлар қиймати: 55.2 сўм.

62.Фойдаланувчининг оптимал вазнини ҳисобловчи, уни реал вазн билан таққословчи ва фойдаланувчига семириш ёки озиш зарурияти тўғрисида тавсиялар берувчи дастур тузинг. Оптимал вазн бўй узунлигидан (см) 100 ни айириш натижасига тенг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Битта сатрдан пробел орқали бўй (см) ва вазн (кг) қийматларини киритинг ва <Enter> ни босинг → **170 68**

Сиз 2 кг семиришингиз керак.

- 63.** 1 дан 100 гача бўлган сонларнинг жуфтларини квадратга, тоқларини куб даражага кўтариб, уларнинг йиғиндисини ҳисобланг.

2.1.7. CASE КЎРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- **Case** кўрсатма дастур кейинги йўлининг (бажарилиши керак бўладиган кўрсатмалар кетма-кетлигида) бир нечта йўналишларидан бирини танлаш керак бўлганда ишлатилади.
- Кетма-кет кўрсатмалардан бирини танлаш дастур бажарилиши вақтида, ўзгарувчининг кўрсатмалар гуруҳи бошида берилган сараловчи ўзгармас қийматига тенглигига боғлиқ равишда амалга оширилади.
- Сараловчи ўзгарувчи сифатида бутун (**integer**) ёки символ (**char**) туридаги ўзгарувчилардан фойдаланилади.

МАСАЛАЛАР

- 64.** Фойдаланувчидан ой рақамини сўровчи ва бу ой йилнинг қайси мавсумига тегишли эканлиги тўғрисидаги жавоб берувчи дастур тузинг. Агар фойдаланувчи мумкин бўлмаган сон киритса, дастур хато сон берилганлиги тўғрисида ахборот чиқарсин. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Ой рақамини киритинг (1 дан 12 гача) ва <Enter> ни босинг → **11**

Қиш

- 65.** Шаҳарлараро телефон орқали сўзлашув қийматини ҳисобловчи дастур тузинг (у минут баҳоси - абонент турининг шаҳаргача бўлган масофаси - узунлиги билан аниқланади). Дастур учун шаҳар коди ва сўзлашув вақти бошланғич қийматлар бўлиб хизмат қилади. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Телефон орқали сўзлашувни ҳисоблаш.

Бошланғич қийматларни киритинг:

Шаҳар коди → **812**

Сўзлашув вақти (бутун сон минут) → **3**

Шаҳар: Тошкент

Минут баҳоси: **150** сўм

Сўзлашув вақти: **450** сўм

Шаҳар	Код	Бир минут баҳоси, сўм
Тошкент	371	150
Самарқанд	322	120
Қарши	375	100
Бухоро	365	130

66. Кейинги кун санасини ҳисобловчи дастур тузинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Бугунги санани рақамларда киритинг (сон ой йил) → **31 12 2005**.

Ойнинг охирги куни!

Келаётган янги йил билан!

Эртага 1.1.2006.

67. Фойдаланувчидан ҳафта куни рақамини сўровчи ва шу куннинг номланишини экранга чиқарувчи дастур тузинг.

2.1.8. FOR КўРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- **for** кўрсатма дастурда, ишлаб чиқиш вақтида аниқланувчи, такрорлашларнинг маълум сондаги цикллари ташкил этиш учун ишлатилади.
- Такрорлашлар сони ўзгарувчи – ҳисобчининг бошланғич ва охирги қийматлари билан аниқланади.
- Ўзгарувчи – ҳисобчи бутун турда (*integer*) бўлиши керак.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

68. Биринчи ўнта бутун мусбат сонлар квадратлари жадвалини чиқарувчи дастур тузинг. Қуйида дастур ишлаш вақтида ҳосил бўлиши керак бўладиган экран кўриниши тавсия этилади:

Квадратлар жадвали

Сон	Квадрат
1	1
2	4
3	9
4	16
5	25

Сон	Квадрат
6	36
7	49
8	64
9	81
10	100

69.Биринчи «n» та бутун мусбат сонлар йиғиндисини ҳисобловчи дастурни ёзинг. Қўшиладиган сонлар дастур ишлаётган вақтда киритилиши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Мусбат сонлар йиғиндисини ҳисоблаш.

Қўшиладиган сонлар миқдорини киритинг → **20**

Биринчи 20 та мусбат сонлар йиғиндисини: 210 га тенг.

70.Биринчи бутун мусбат тоқ сонлар квадратлари жадвалини чиқарувчи дастур тузинг.

71.Дастлабки n та бутун мусбат сонлар йиғиндисини ҳисобловчи дастур тузинг.

72.1,3,5,7, ... қаторнинг биринчи n та ҳади йиғиндисини ҳисоблаш дастурини тузинг. Қўшилувчи ҳадлар сони дастур иши вақтида берилади. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

1,3,5,7, ... қаторнинг хусусий йиғиндисини топиш.

Қаторнинг қўшилувчи ҳадлар сонини киритинг → **15**

Қаторнинг дастлабки 15 та ҳади йиғиндисини: 330.

73. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ қаторнинг биринчи n та ҳади йиғиндисини ҳисоблаш дастурини тузинг. Қўшилувчи ҳадлар сони дастур иши вақтида берилади. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ қатор хусусий йиғиндисини ҳисоблаш.

Қатор қўшилувчи ҳадлар сонини киритинг → **15**

Қаторнинг дастлабки 15 та ҳади йиғиндисини: 3.3182

74. Икки сонининг 0 дан 10 гача бўлган даражалари жадвалини чиқарувчи дастур тузинг. Қуйида дастурнинг иш вақтида тавсия этиладиган экран кўриниши келтирилади.

Иккининг даражалари	Натижа
0	1
1	2
2	4
3	8
4	16
5	32

Иккининг даражалари	Натижа
6	64
7	128
8	256
9	512
10	1024

75. Соннинг факториалини ҳисобловчи дастур тузинг.

76. Клавиатурадан киритилган 5 та каср сонларнинг ўрта қийматини ҳисобловчи дастур тузинг.

77. Клавиатурадан киритилган сонлардан 1 дан 10 гача бўлганларини такрорлаб олиб, уларнинг ўрта арифметик қийматини ҳисобловчи дастур тузинг.

78. Маҳсулотнинг оғирлигига кўра баҳоси жадвалини экранга чиқарувчи жадвал тузинг. Масалан 1 кг 16 сўм 50 тийин бўлсин.

79. $y = -2,4x^2 + 5x - 3$ функциянинг -2 дан 2 гача соҳада 0,5 қадам билан ўзгаргандаги қийматлари жадвалини чиқарувчи дастурни тузинг. Қуйида дастурнинг иш вақтида тавсия этиладиган экран кўриниши келтирилади.

X	Y
- 2	-22,60
- 1,5	-15,90
- 1	-10,40
- 0,5	-6,10
0	-3,00

X	Y
0,5	-1,10
1	0,40
1,5	-0,90
2	-2,60

80. $y = |x + 2|$ функциянинг қийматлар жадвалини экранга чиқарувчи дастур тузинг. «x» -4 дан 4 гача 0,5 қадам билан ўзгаради.

81. $y = |x - 2| + |x + 1|$ функциянинг қийматлар жадвалини экранга чиқарувчи дастур тузинг. «х» -4 дан 4 гача 0,5 кадам билан ўзгаради.

82.Клавиатурадан бешта каср сондан иборат кетма-кетликни киритиш дастурини тузинг. Дастурда ҳар бир сон киритилгандан кейин кетма-кетликнинг ҳосил бўлган қисми ўрта арифметик қийматини чиқариш ҳисобга олинган бўлсин. Дастурнинг иш вақтида тавсия этиладиган экран кўриниши қуйида келтирилган.

Каср сонлар кетма-кетлигини қайта ишлаш.

Ҳар бир сон киритилгандан кейин <Enter> ни босинг → **12.3**

Киритилган сон: 1 та йиғинди 12.30 ўрта арифметик: 12.30 → **15**

Киритилган сон: 2 та йиғинди 27.30 ўрта арифметик: 13.65 → **10**

Киритилган сон: 3 та йиғинди 37.30 ўрта арифметик: 12.43 → **5.6**

Киритилган сон: 4 та йиғинди 42.90 ўрта арифметик: 10.73 → **11.5**

Киритилган сон: 5 та йиғинди 54,40 ўрта арифметик: 10,88

Дастур ишини тугатиш учун <Enter> ни босинг.

83. Клавиатурадан киритилаётган каср сонлар кетма-кетлигининг ўрта арифметигини ҳисоблаш дастурини тузинг. Охириги сон киритилгандан кейин дастур кетма-кетлигининг энг кичик ва энг катта сонини чиқариши керак. Сонлар миқдори дастур иши вақтида берилади. Қуйида дастурнинг иш вақтида тавсия этиладиган экран кўриниши келтирилган.

Каср сонлар кетма-кетлигини қайта ишлаш.

Кетма-кетликдаги сонлар миқдорини киритинг → **5**

Кетма-кетликни киритинг. Ҳар бир сон киритилгандан кейин <Enter> ни босинг.

→ **5.4**

→ **7.8**

→ **3.0**

→ **1.5**

→ **2.3**

Сонлар миқдори: 5

Ўрта арифметик: 4.00

Минимал сон: 1.5

Максимал сон: 7.8

Дастурнинг ишини тугатиш учун <Enter> клавишини босинг.

84.1 дан 10 гача соҳадаги 10 та тасодифий сондан иборат учта кетма-кетликни ажратувчи дастурни ёзинг. Дастур ҳар бир кетма-кетликни экранга чиқаради ва ҳар бир кетма-кетлик ўрта арифметигини ҳисоблайди. Дастурнинг ишлаш вақтидаги экран кўриниши қуйида келтирилган.

*** Тасодифий сонлар ***

6	10	4	2	5	8	1	7	7	3	Ўрта арифметиги 5.30
10	3	6	1	10	1	3	8	7	6	Ўрта арифметиги 5.50
5	2	2	5	4	2	2	1	6	10	Ўрта арифметиги 3.90

Ишни тугатиш учун <Enter> клавишини босинг.

85. $y=|x|$ функциянинг аргумент соҳаси -4 дан 4 гача 0,5 қадам билан ўзгаргандаги қийматлари жадвалини ҳосил қилиш дастурини тузинг

86.Экранга кўпайтириш (масалан, 7 га) жадвалини чиқариш дастурини тузинг. Дастур ишлаш вақтида тавсия қилинадиган экран кўриниши қуйидагича:

$7 \times 2 = 14$
 $7 \times 3 = 21$
 $7 \times 4 = 28$
 $7 \times 5 = 35$
 $7 \times 6 = 42$
 $7 \times 7 = 49$
 $7 \times 8 = 56$
 $7 \times 9 = 63$

87.Клавиатурадан 0 дан 255 гача бўлган соҳадаги ўнлик бутун сонни иккиликда бериш дастурини ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Ўнлик сонни иккиликка айлантириш.

[0; 255] соҳадан бутун сон киритинг ва <Enter> ни босинг → **164**

Ўнлик 164 сонига 10100100 иккилик сони мос келади.

Дастур ишини тугатиш учун <Enter> клавишини босинг.

88.Экранга Пифагор квадратини - кўпайтириш жадвалини - чиқариш дастурини тузинг. Дастур ишлаш вақтида тавсия қилинадиган экран кўриниши қуйидагича:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90

89. $1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$ қаторнинг хусусий йиғиндисини ҳисоблайдиган ва ҳосил бўлган қийматни $\pi/4$ билан таққослайдиган (етарли даражада катта миқдордаги ҳадлар қўшилганда, хусусий йиғинди $\pi/4$ га яқинлашиб боради) дастур тузинг.

90. Кўпайтириш жадвалини билишни текширувчи дастур тузинг. Дастур бўйича 10 та мисол ишланиши, 10 та тўғри жавоб учун «аъло», 9 ва 8 таси учун «яхши», 7 таси учун «қониқарли», 6 ва ундан кам тўғри жавоблар учун «ёмон» баҳо қўйилиши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

$$5 \times 3 = \mathbf{15}$$

$$7 \times 7 = \mathbf{49}$$

$$1 \times 4 = \mathbf{4}$$

$$4 \times 3 = \mathbf{12}$$

$$9 \times 4 = \mathbf{36}$$

$$8 \times 8 = \mathbf{64}$$

$$7 \times 8 = \mathbf{52}$$

$$\text{Адашдингиз! } 7 \times 8 = 56$$

$$4 \times 7 = \mathbf{28}$$

$$3 \times 5 = \mathbf{15}$$

$$2 \times 5 = \mathbf{10}$$

Тўғри жавоблар: 9

Баҳо: яхши

91. 100 гача бўлган сонларни кўшиш ва айиришни бажаришни текшириш. Дастур бўйича 10 та мисол ишланиши керак, ҳар бир мисолда камаювчи сон айирилувчи сондан катта ёки унга тенг бўлиши керак (манфий сон

ҳосил қилиш мумкин эмас). Баҳолаш қуйидагича бўлади: 10 та тўғри жавоб учун «аъло», 9 ва 8 таси учун «яхши», 7 таси учун «қониқарли», 6 ва ундан кам тўғри жавоблар учун «ёмон» баҳо. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Сонларни қўшиш ва айириш кўникмасини текшириш.

Мисолдан кейин жавобни киритинг ва <Enter> ни босинг.

$$75 - 4 = 71$$

$$35 - 9 = 29$$

Адашдингиз! $35 - 9 = 26$

$$14 - 1 = 13$$

$$6 - 5 = 1$$

$$37 - 19 = 28$$

Адашдингиз! $37 - 19 = 18$

$$53 - 14 = 39$$

$$94 - 87 = 7$$

$$90 - 16 = 74$$

$$4 - 2 = 2$$

$$89 - 41 = 48$$

Тўғри жавоблар: 8

Баҳо: яхши

- 92.**Экранга, масалан, уч минут мобайнида ёки фойдаланувчи ихтиёрий клавишни босгунга қадар ишлайдиган «электрон соат» ни чиқарувчи дастурни тузинг.
- 93.**Экранга шахмат тахтаси ифодасини чиқарувчи дастурни тузинг. Юлдузчалар (*) қора катакларни, бўшлиқ оқ катакларни билдирсин.
- 94.**«m» ва «n» ўзгарувчилари оралиғида қийматлар квадратлари йиғиндисини ҳисобланг.
- 95.** $a \leq 20$; $1 \leq b \leq 20$ бўлганда $a^2 + b^2 = c^2$ тенгликни қаноатлантирувчи (Пифагор) сонларини аниқланг.
- 96.**Рақамлар кубларининг йиғиндиси соннинг ўзига тенг бўладиган барча уч хонали сонларни топинг (масалан, $407 = 4^3 + 0^3 + 7^3 = 64 + 0 + 343$).

2.1.9. REPEAT КўРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- циклнинг **repeat** кўрсатмаси такрорланишлар сонини дастур бажарилишининг бориши билан аниқлаб боради;
- **repeat** цикл кўрсатмаси **until** сўзидан кейинги шарт ёлгон бўлгунча бажарилади;
- **until** сўзидан кейин циклнинг тугаш шартини ёзиш керак;
- **repeat** циклини тугатиш учун цикл танасида циклнинг тугашига таъсир қилувчи кўрсатма, албатта, бўлиши ва бажарилиши керак;
- **repeat** циклида цикл танасидаги кўрсатмалар ҳеч бўлмаганда бир марта бажарилади;
- **repeat** цикли, одатда, тақрибий ҳисоблашларни ташкил қилишда, излаш ва клавиатура ёки файлдан киритиладиган маълумотларни қайта ишлашда ишлатилади.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган.

МАСАЛАЛАР

97.Фойдаланувчи киритган бутун сон оддий эканлигини текшириш дастурини тузинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Бутун сонни киритинг ва **<Enter>** ни босинг.

→**45**

45 – оддий сон эмас.

98.1 дан 10 гача бўлган соҳадаги сон «ўйлайдиган» ва фойдаланувчига 5 та уринишда сонни топадиган дастур тузинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

«Сонни топ» ўйини

Компьютер 1 дан 10 гача бўлган сон «ўйлади»

Уни 5 уринишда топинг

Сонни киритинг ва **<Enter>** ни босинг.

→ **5**

Йўқ

→ **3**

Сиз ютдингиз! Табриклайман!

99.Берилган вақт оралиғидан кейин овоз сигнали берувчи дастур – таймер тузинг. Вақт интервали миқдори дастур иши вақтида клавиатурадан киритилади.

- 100.** Клавиатурадан киритиладиган мусбат сонлар ўрта арифметигини топиш дастурини ёзинг.
- 101.** Клавиатурадан кетма-кет киритилган сонлардан максималини топиш дастурини ёзинг.
- 102.** Интегрални трапеция усули билан тақрибий ҳисоблаш дастурини ёзинг. Циклнинг ҳар бир такрорланишида дастур интервал миқдорлари ва катталигини чиқариб борсин.

2.1.10. WHILE КўРСАТМАСИ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- **while** сўзидан кейин ёзилган шарт ҳақиқат бўлгунча while циклининг кўрсатмалари бажарилади;
- **while** сўзидан кейин цикл кўрсатмасининг бажарилиш шартини ёзиш керак;
- **while** циклини тугатиш учун цикл танасида цикл кўрсатмасининг бажарилиш шартига таъсир этувчи кўрсатмалар бўлиши шарт;
- **while** цикли баъзи ҳолларда бир марта ҳам бажарилмаслиги мумкин;
- **while** цикли, одатда, тақрибий ҳисоблашларни ташкил қилишда, излаш ва клавиатура ёки файлдан киритиладиган маълумотларни қайта ишлашда ишлатилади.

Изоҳ: масала шартларида дастурчи киритилган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

- 103.** Фойдаланувчи берган аниқлик билан π ни ҳисоблаш дастурини ёзинг. Бунинг учун қуйидагидан фойдаланинг: $1-1/3+1/5-1/7+1/9-\dots$ қаторнинг хусусий йиғиндиси етарли даражада катта сондаги ҳадларни қўшганда $\pi/4$ га яқинлашиб келади. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

π ҳисоблаш аниқлигини беринг → **0.001**

π сонининг 0.001000 аниқлик билан қиймати → **3.143589**

Қаторнинг 502 ҳади қўшилган.

- 104.** Иккита бутун соннинг энг катта умумий бўлувчисини топиш дастурини ёзинг.

105. a сонини b сонига бўлишдаги қолдиғини, ($a \geq 0$; $b > 0$) DIV ва MOD амалларидан фойдаланмай туриб, аниқланг.
106. $e = a^4$ ни даражага кўтариш амалини бажармасдан ҳисобланг.
107. 100 сонигача бўлган жуфт сонларнинг кублари ва тоқ сонларнинг квадратлари йиғиндисини ҳисобланг.
108. Берилган туб сонни кўпайтувчиларга ажратинг.

2.1.11. БЕЛГИЛАР ВА ҚАТОРЛАР

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- ҳар бир белгига сон – белги коди мос келади;
- сатрнинг алоҳида белгисига йўл топиш учун ўзгарувчи (сатрнинг) номидан кейин квадрат қавсларда унинг рақамини кўрсатиш мумкин.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган.

МАСАЛАЛАР

109. Фойдаланувчидан унинг номини ва соғлиғини сўровчи дастур ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Исмингиз нима?

Исмингизни киритинг ва **<Enter>** ни босинг

→ **Эшмат**

Ассалому алайкум, Эшмат!

110. «Телеграф» йўсинида (яъни ахборот ҳарфлари биттадан, баъзи бир сукут билан пайдо бўлиши керак) экранга ахборот чиқарувчи дастурни ёзинг.

111. Фойдаланувчи томонидан киритилган белги кодини ASCII стандартида чиқарувчи дастурни ёзинг. Дастур ишини, масалан, нукта киритилгандан кейин тугатиши керак. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Белгини киритинг ва **<Enter>** ни босинг.

Дастур ишини тугатиш учун нуктани киритинг → •

→ **1**

Белги: 1 код: 81

→ 2

Белги: 8 код: 88

→ E

Белги: E код: 101

→ •

112. Белгилар коди жадвалининг биринчи қисмини (0 дан 127 гача кодга эга бўлган белгилар) экранга чиқариш дастурини ёзинг. Жадвал саккизта устун ва ўн олти сатрдан иборат бўлиши керак. Биринчи устунда 0 дан 15 гача кодга эга бўлган белгилар, иккинчисидан 16 дан 31 гача ва ҳ.к бўлиши керак.

113. Клавиатурадан киритилган рус алфавитининг ёзма ҳарфларини босма ҳарфларга айлантириш дастурини ёзинг (Рус алфавитли UpCase функцияси ишламаслигини эътиборга олинг). Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Матн сатрини киритинг ва <Enter> ни босинг.

→ *Паскални ўргангач, Дельфига ўтиш мумкин.*

Юқори регистрда ўзгартирилган сатр:

ПАСКАЛНИ ЎРГАНГАЧ, ДЕЛЬФИГА ЎТИШ МУМКИН.

114. Клавиатурадан киритилган сатрнинг бошланғич бўш хоналарини олиб ташловчи дастурни ёзинг.

115. Клавиатурадан киритилган сатр бутун сон эканлигини текширувчи дастурни ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Сонни киритинг ва <Enter> ни босинг.

→ 23.5

Киритилган сатр бутун сон эмас.

116. Клавиатурадан киритилган сатр ўн олтилик сон эканлигини текшириш дастурини ёзинг.

117. Клавиатурадан киритилган сатр каср сон эканлигини текшириш дастурини ёзинг.

118. Клавиатурадан киритилган саккиз хонали иккилик сонни ўнликка айлантириш дастурини ёзинг. Қуйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Саккиз хонали иккилик сонни киритинг ва <Enter> ни босинг.

→ 11101010

Иккилик 11101010 сонига ўнлик 324 сони мос келади.

Дастур ишини тугатиш учун <Enter> ни босинг.

119. Клавиатурадан киритилган икки хонали ўн олтилик сонни ўнликка айлантириш дастурини ёзинг.

120. Фойдаланувчи томонидан киритилган ўнлик сонни кўрсатилган санок системасидаги (2 ликдан 10 ликкача) сонга ўтказиш дастурини ёзинг. Дастур ишлаш вақтида экраннинг тавсия қилинадиган кўриниши куйидагича:

Бутун сонни киритинг → **67**

Санок системаси асосини киритинг → **2**

100011

121. Фойдаланувчи томонидан киритилган ўнлик сонни ўн олтиликка ўтказиш дастурини ёзинг.

122. $N_0O_1, N_1O_2, \dots, O_kN_k$ ифода қийматини ҳисобловчи дастурни ёзинг, бу ерда N_i – бутун бир хонали сон, O_i – оддий арифметик амалларнинг (кўшиш ёки айтириш) икки белгисидан бири. Куйида дастур ишлаш вақтида тавсия қилинадиган экран кўриниши берилган.

Арифметик ифодани киритинг, масалан, $4 + 5 - 3 - 5 + 2$ ва <Enter> ни босинг.

→ **9 - 5 + 4 + 2 - 6**

Киритилган ифода қиймати: 4

Дастур ишини якунлаш учун <Enter> ни босинг.

2.1.12. МАССИВЛАР

Бу бўлим масалаларини ечишга ўтишдан олдин куйидагиларни эслаш керак:

- Массив бу бир хил турдаги элементлар тупламини ифодаловчи маълумотлар таснифидир;
- Массив ўзгарувчиларни эълон қилиш бўлимида эълон қилиниши керак;
- Массивларни эълон қилишда номланган ўзгармаслардан фойдаланиш кулай;
- Массив, элементига йўл элемент индекси (рақами) ни кўрсатиш орқали очилади. Индекс сифатида бутун турдаги ифода, масалан, бутун ўзгармас ёки **integer** туридаги ўзгарувчи ишлатилади;

- Массивларни киритиш, чиқариш ва қайта ишлашда цикл кўрсатмаларидан (**for**, **while**, **repeat**) фойдаланиш қулай.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

123. Клавиатурадан бешта бутун сондан иборат бир ўлчовли массивни киритувчи ва нолмас элементлар миқдорини ҳисобловчи дастурни ёзинг. Ҳар бир элементни киритишдан олдин экранда тартиб рақами билан белги изоҳи чиқсин.

Ҳар бир сон киритилгач, **<Enter>** босилсин.

a [1] → **12**

a [2] → **0**

a [3] → **3**

a [4] → **-1**

a [5] → **0**

Массивда нолмас элементлар учта

124. Клавиатурадан киритилган бутун сонлар массивидан энг кичик қийматлисини топиш дастурини тузинг. Қуйида иш вақтида тавсия этилувчи, экран кўриниши берилган.

Массивнинг минимал элементини топиш.

Бир сатрдан массив элементларини (5 та бутун сон) киритинг ва **<Enter>**ни босинг → **23 0 45 -5 12**

Массивнинг минимал элементи: **-5**

125. Куннинг ҳафталик ўртача ҳароратини ҳисобловчи дастурни ёзинг. Бошланғич қийматлар дастур ишлаш вақтида киритилиши мумкин. Қуйида иш вақтида тавсия этилувчи экран кўриниши берилган.

Ҳафталик ҳаво ҳарорати маълумотларини киритинг.

Душанба → **12**

Сешанба → **10**

Чоршанба → **16**

Пайшанба → **18**

Жума → **17**

Шанба → **16**

Якшанба → **14**

Ҳафталик ўртача ҳарорат; 14.71 град.

126. Массивда клавиатурадан киритилган сон бор-йўқлигини текширувчи дастурни ёзинг. Ишлаш вақтида массив киритилиши керак.
127. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Унинг ҳамма манфий қийматларини квадратлари билан алмаштиринг.
128. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Шу вектор элементларидан (с,d) оралиқда етган элементлар йиғиндиси ва миқдорини топинг.
129. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор ва X ҳақиқий сон берилган. Вектор элементларини тартибга солинг ва соннинг элементларнинг қайси оралиғида ётишини аниқланг.
130. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор ва X ҳақиқий қийматли сон берилган. Вектор элементларидан нечтаси шу сонга тенг бўлишини аниқланг.
131. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Шу векторнинг мусбат элементларини 0,5 га камайтириб боринг.
132. Массивнинг клавиатурадан киритилган элементларини ўсиб бориш кетма-кетлигида берилганини текширувчи дастурни ёзинг.
133. Клавиатурадан киритилган сон массивда нечанчи ўринда туришини аниқловчи дастурни ёзинг.
134. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Шу векторда нечта мусбат, нечта манфий сон ва нечта ноль борлигини аниқланг.
135. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Векторнинг манфий элементигача бўлган элементлар миқдорини ҳисобланг.
136. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. «Y» ўзгарувчига a векторнинг ҳамма мусбат элементлари йиғиндисининг иккиланган қийматини ўзлаштиринг.
137. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Вектор тоқ рақамли элементларини 5 га, жуфт рақамли элементларини 3 га кўпайтириб, уларнинг йиғиндисини топинг.
138. $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Унинг манфий элементларини (векторда жойлашиш тартибини бузмасдан) биринчи, манфий бўлмаган элементларини эса кейинги ўринларда жойлаштириб, янги $b\{b_1, b_2, \dots, b_n\}$ вектор ҳосил қилинг.

139. Клавиатурадан киритилган бир ўлчовли массивни тўғри танлов усули билан камайиш тарзида сараловчи дастурни ёзинг.

140. Клавиатурадан киритилган бир ўлчовли массивни алмаштириш («кўпик») усули билан камайиш тарзида сараловчи дастурни ёзинг.

141. Ўсиб бориш тарзида тартибга солинган 2 массивни бирлаштирувчи дастурни ёзинг. Қуйида иш вақтида тавсия этилувчи экран кўриниши берилган.

Тартибга солинган иккита массивни бирлаштириш.

Битта сатрдан биринчи массив элементларини киритинг (5 та бутун сон) → **1 3 5 7 9**

Битта сатрдан иккинчи массив элементларини киритинг (5 та бутун сон) → **2 4 6 8 10**

Натижа – массив

1 2 3 4 5 6 7 8 9 10

Ишни якунлаш учун <**Enter**>ни босинг.

142. Бинар излаш усулини қўллаб, ўсиб бориш тарзида тартибга солинган массивда излашни амалга оширувчи дастурни ёзинг.

143. Гуруҳда бўйи ўртачадан баланд бўлган талабалар сонини аниқлаш дастурини тузинг. Қуйида иш вақтида тавсия этилувчи экран кўриниши берилган

*** Талабалар бўйи таҳлили ***

Бўйини киритинг (см) ва <**Enter**> ни босинг.

Ишни якунлаш учун 0 киритинг ва <**Enter**> ни босинг.

→ **175**

→ **170**

→ **180**

→ **168**

→ **170**

→ **0**

Ўртача бўй: 172.6 см.

Иккита талабада бўйи ўртачадан баланд.

144. Клавиатурадан сатрлар бўйича икки ўлчовли массивни киритувчи ва унинг устунлар бўйича йиғиндисини топувчи дастурни ёзинг.

145. Иккинчи тартибли квадрат матрица аниқловчисуё ҳисобловчи дастурни ёзинг. Қуйида иш вақтида тавсия этилувчи экран кўриниши берилган.

Иккинчи тартибли матрицани киритинг.

Сатр элементлари киритилгач, **<Enter>** ни босинг.

→ **5 -7**

→ **1 3**

Матрица аниқловчиси:

5.00 -7.00

1.00 3.00

22.00 га тенг.

146. Клавиатурадан киритилган квадрат матрица «афсункор» (магик) квадрат эканлигини текширинг.

«Афсункор» квадрат деб, ҳар бир сатр, ҳар бир устун ва ҳар бир диагонал бўйича йиғиндиси бир хил сонга тенг матрицага айтилади (қуйига қаранг).

2	9	4
7	5	3
6	1	8

13	8	12	1
2	11	7	14
3	10	6	15
16	5	9	4

147. Универсиада ўйинлари натижаларини яқунловчи дастур ёзинг. Фойдаланувчи дастурга ҳар бир иштирокчи – гуруҳ қўлга киритган ҳар хил даражадаги медаллар сонини киритиши, дастур эса медалларнинг умумий сони ва уларга мос келувчи очколарни ҳисоблаши, шундан кейин тўпланган очколар сонига мос равишда (олтин медал учун – 7, кумуш учун – 6, бронза учун – 5 очко олади) рўйхатни тартибга солиши керак. Дастурнинг ишлаш вақтида экраннинг тавсия этилувчи кўриниши қуйида келтирилган.

Универсиада ўйинлари натижаси.

Бир сатрда олтин, кумуш ва бронза медаллари сонини киритилган.

Андижон →	3	5	9
Тошкент ш.→	12	9	8
Жиззах →	6	5	4
Навоий →	0	6	2
Наманган →	3	1	2
Бухоро →	10	10	5
Сурхондарё →	9	6	3
Самарканд →	6	3	4
Сирдарё →	2	2	1
Фаргона →	5	1	4
Хоразм →	3	5	10
ККАССР →	3	4	5
Қашқадарё →	2	6	9
Тошкент →	2	4	6

Универсиада ўйинлари яқунлари

	Мамлакат	Олтин	Кумуш	Бронза	Жами	Очколар
1.	Тошкент ш	12	9	8	29	178
2.	Бухоро	10	10	5	25	155
3.	Сурхондарё	9	6	3	18	114
4.	Андижон	3	6	9	18	102
5.	Хоразм	3	5	10	18	101
6.	Қашқадарё	2	6	9	17	95
7.	Жиззах	6	5	4	15	92
8.	Самарканд	6	3	4	13	80

9.	ҚҚАССР	3	4	5	12	70
10.	Сирдарё	2	4	6	12	68
11.	Фарғона	5	1	4	10	61
12.	Навоий	0	6	2	8	46
13.	Наманган	3	1	2	6	37
14.	Тошкент в.	2	2	1	5	31

147. «Сонни топ» ўйин дастурини тузинг. Ўйин қоидалари куйидагича. Икки киши ўйнайди. Бири сон ўйлайди, иккинчиси уни топади. Ҳар кадамда топувчи-ўйинчи ўз фаразини айтади, сонни ўйлаган эса соннинг нечта рақами топилганини ва бу рақамлардан нечтаси сонда тўғри ўринни эгаллаганини айтади. Масалан, агар 725 сони ўйланган ва 523 тахмин киритилган бўлса, иккита (5 ва 2) сони топилган ва улардан биттаси (2) тўғри ўринни эгаллаган, деб ҳисобланади. Қуйида иш вақтида тавсия этилувчи экран кўриниши берилган.

Компьютер уч хонали сон ўйлади. Сиз уни топишингиз керак. Навбатдаги сон киритилгандан кейин Сизга нечта рақам топилгани ва улардан нечтаси ўз ўринларида эканлиги айтилади.

Сонни киритгандан кейин <Enter> ни босинг.

Ўйинни тугатиш учун <Esc> ни босинг.

Сизнинг вариантингиз → 123	Топилди: 0.	Ўз ўрнида: 0
Сизнинг вариантингиз → 456	Топилди: 1.	Ўз ўрнида: 0
Сизнинг вариантингиз → 654	Топилди: 2.	Ўз ўрнида: 2
Сизнинг вариантингиз → 657	Топилди: 2.	Ўз ўрнида: 2
Сизнинг вариантингиз → 658	Топилди: 3.	Ўз ўрнида: 3

*** Сиз топдингиз ! ***

Ишни тугатиш учун <Enter> ни босинг!

148. Фойдаланувчидан ахборот қабул қилувчи ва уни экранга нуқта ва тирелар кетма-кетлиги кўринишида чиқарувчи телеграф-дастур ёзинг. Нуқта ва тиреларни, мос узунликда, овоз сигналлари ҳамроҳлигида бериш мумкин. Рус харфларининг Морзе Алфавити куйида келтирилган.

А	.-	Б	-. . . .	В	. - - -	Г	- - .
Д	- . .	Е	. - . - .	Ж	. . . -	З	- - . . .
И	. .	Й	. - - -	К	- . -	Л	. - . .
М	- -	Н	- .	О	- - -	П	. - - .
Р	. - .	С	. . .	Т	-	У	. . -
Ф	Х	Ц	- . . .	Ч	- - - .
Ш	- - - -	Щ	- - . -	Ъ	- . . -	Ы	- . - -
Ь	- . . -	Э	Ю	. . - -	Я	. - . -

2.1.13. ФУНКЦИЯЛАР

Бу бўлим масалаларини ечишга ўтишдан олдин куйидагиларни эслаш керак:

- маълумотларни функцияга узатишдан олдин фақат параметрларни (глобал ўзгарувчилар, яъни функциядан ташқарида эълон қилинган ўзгарувчиларни ишлатиш тавсия этилмайди) ишлатиш керак;
- функциянинг бажарилувчи кўрсатмалари ичида функция қийматларини ўзлаштириш кўрсатмалари албатта бўлиши керак;
- функция чақириш кўрсатмасида ҳар бир ҳақиқий параметр тури (ўзгармаслар ёки ўзгарувчи) функцияни эълон қилишда кўрсатилган мос расмий параметр билан бир хил бўлиши керак;
- агар функцияни эълон қилиш кўрсатмасида расмий параметр исми олдида **var** сўзи бўлмаса, функцияни чақириш вақтида расмий параметр сифатида ўзгармас ёки мос турдаги ўзгарувчини ишлатиш мумкин. Агар **var** сўзи кўрсатилган бўлса, расмий параметр сифатида фақат ўзгарувчи бўлиши мумкин.
- Агар функция параметрлари натижани фақат функцияни чақирган дастурга қайтариш учун ишлатилса, функцияни эълон қилишда тегишли параметр исми олдида **var** сўзи бўлиши керак.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

- 149.** Аргумент сифатида олинган иккита бутун сондан каттасини қайтарувчи функцияни ёзинг.
- 150.** Иккита бутун сонни таққословчи ва таққослаш натижасини $>$, $<$ ёки $=$ белгилардан бири кўринишида қайтарувчи функцияни ёзинг.
- 151.** Иккита резистордан иборат занжир қаршилигини ҳисобловчи функцияни ёзинг. Қаршилик миқдорлари ва улаш турлари (кетма-кет ёки параллел) функция параметрлари бўлади. Функция параметрларнинг тўғрилигини текшириши керак: агар улаш тури хато кўрсатилган бўлса, функция 1 ни қайтариши керак.
- 152.** a^b қийматни ҳисобловчи функцияни ёзинг, a ва b сонлар ихтиёрий каср мусбат сонлар бўлиши мумкин.
- 153.** Омонат бўйича даромадни ҳисобловчи **Daromad** функциясини ёзинг. Омонат миқдори, йиллик фоиз устама ва омонат муддати (кунлар сони) функциянинг бошланғич маълумотлари бўлади.
- 154.** Агар аргумент сифатида функция томонидан қабул қилинган белги алфавитнинг ўнли ҳарфи бўлса, **True** қийматини қайтарувчи **Unli** функциясини ёзинг.
- 155.** Аргумент сифатида олинган сатрдаги бошланғич бўшлиқ (пробел) ларни олиб ташловчи **LTrim** функциясини ёзинг.
- 156.** Аргумент сифатида ҳосил қилинган ўзгартирилган сатрни юқори регистрга қайтарувчи функцияни ёзинг. Эътибор беринг, агар бу функция **UpCase** (лотин алфавити белгилари учун шунга ўхшаш масала ечувчи функция Turbo Pascalда шундай деб аталади) деб аталса, дастурчининг функцияси Turbo Pascal функцияси ўрнини босади.
- 157.** Квадрат тенгламани ечувчи функцияни ёзинг. Тенглама коэффициентлари ва илдизлари функция параметрлари бўлиши керак. Функция қиймати уни чақирувчи дастурга тенглама илдизи борлиги ҳақидаги ахборотларни узатиш учун ишлатилади: 2-ҳар хил илдизлар, 1-бир хил илдизлар, 0-тенглама ечимга эга эмас. Бундан ташқари, функция бошланғич маълумотларнинг тўғрилигини текшириши керак. Агар бошланғич маълумотлар хато бўлса, функция 1 ни қайтаради.

- 158.** Клавиатурадан бутун икки хонали мусбат сонни киритишни таъминловчи функцияни ёзинг. Фойдаланувчи қандайдир бир клавишни босиши билан унга мос келувчи белги экранда, у фақат рақам бўлган ҳолдагина, пайдо бўлиши керак. Функция фойдаланувчига киритилган сонни <BackSpace> клавиши ёрдамида таҳрирлашга имкон бериши керак. Агар фойдаланувчи ҳеч бўлмаганда битта рақамни киритган бўлса, <Enter> клавишининг босилиши билан функция ишини тугатиш ва фойдаланувчи киритган сонни уни чақирган процедурага қайтариши керак.
- 159.** Цилиндр ҳажмини ҳисобловчи функцияни ёзинг. Функция параметрлари цилиндр радиуси ва баландлиги.
- 160.** Аргумент сифатида функция томонидан қабул қилинган рус алфавитидаги ҳарф ундош бўлганда, TRUE қийматини берувчи Undosh функциясини ёзинг.
- 161.** Даражага кўтариш функциясини ёзинг.
- 162.** Соннинг факториалини ҳисобловчи функция тузинг ва уни $C_n^m = \frac{n!}{m!(n-m)!}$ ни ҳисоблашда ишлатинг.
- 163.** Сессия натижалари бўйича гуруҳлар ўртача балини топиш дастурини тузинг. Тўртта гуруҳлар бўйича баҳолар қуйидаги матрицаларда келтирилган: A (24,5), B (23,5), C (22,5), D (24,5).
- 164.** A (10,12), B (15,10) ва C (8,10) матрицалар берилган. Уларнинг энг кичик элементларини топиш функциясини тузинг.
- 165.** A (80), B (70), C (100) массивларда келтирилган тажриба натижаларининг ўрта қийматларини функция тузиб, ҳисобланг.
- 166.** X (5,6), Y (7,4), Z (5,7) берилган массивларнинг максимал элементларини топиш функциясини тузинг. Натижаларни P (3) массивда келтиринг.
- 167.** A (40), B (30), C (60) бутун сонли массив элементларининг ичидан 3 га бўлинадиганларини, функция тузиб, топинг.
- 168.** $ax^2 + bx - 4 = 0$; $y^2 - y + d = 0$; $2z^2 + dz - 1 = 0$ квадрат тенгламаларнинг катта илдизларини ҳисоблаш функциясини тузинг ва катта илдизларни B (3) массивга киритинг.

2.1.14. ПРОЦЕДУРАЛАР

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- процедурага ахборотни узатиш учун процедурадан ташқарида эълон қилинган глобал ўзгарувчиларни эмас, балки параметрларни ишлатиш керак;
- процедурани чақириш кўрсатмасидаги ҳар бир ҳақиқий параметр (ўзгармаслар ёки ўзгарувчи) тури функцияни эълон қилишда кўрсатилган тегишли расмий параметрга мос келиши керак;
- агар процедурани эълон қилиш кўрсатмасида расмий параметр исми олдида **var** сўзи бўлмаса, процедурани чақириш вақтида расмий параметр сифатида ўзгармас ёки мос турдаги ўзгарувчини ишлатиш мумкин. Агар **var** сўзи кўрсатилган бўлса, расмий параметр фақат ўзгарувчи бўлиши мумкин.
- агар процедурани чақирган процедура аргументи натижани дастурга қайтариш учун қўлланилса, аргумент номи олдида **var** сўзи ёзилади.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўгон шрифт билан ёзилган

МАСАЛАЛАР

- 169.** Экранга юлдузчалардан иборат сатрни чиқарувчи процедурани ёзинг. Сатр узунлиги (юлдузчалар сони) функция параметри бўлади.
- 170.** Бир хил белгилардан иборат сатрни чиқарувчи процедурани ёзинг. сатр узунлиги ва белги процедура параметлари бўлади.
- 171.** Экранга рамкани чиқарувчи Frame процедурасини ёзинг. процедура параметрлари сифатида рамаанинг юқори чап бурчак координаталари ва унинг ўлчами узатилиши керак.
- 172.** Берилган a ва b сонлардан каттасини X , кичигини Y ўзлаштирадиган дастур тузинг. Масала `min` ва `max` процедураларини ўз ичига олсин.
- 173.** 1 дан n гача бўлган сонларнинг кублари квадратлари йиғиндисини ҳисобловчи процедура тузинг.
- 174.** `Tartib` дастурига иккита ўзгарувчиларнинг қийматларини ўзаро алмаштирадиган процедура тузиб киритинг.
- 175.** Чизиқли тенгламалар системаси илдизларини процедурадан фойдаланиб, Крамер формуласи ёрдамида топинг.

176. $P_n(X) = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X + a_0$ кўпхадни процедурадан фойдаланиб ҳисобланг.
177. $X = X_0, X_1, \dots, X_n$ (бу ерда $X_{i+1} = X_i + h$ ($i=1, 2, \dots, n$)) сонларнинг квадрат ва кубларини ҳисоблашда процедурадан фойдаланинг.
178. $Z_{ij} = X_i Y_j$ ($i=1, 2, \dots, m; j=1, 2, \dots, n$) миқдорни ҳисоблаш процедурасини тузинг. Бу ерда X_i, Y_i мос равишда X ва Y бутун сонли вектор элементлари. Z_{ij} бутун сонли матрица элементлари.
179. $X = \{X_1, X_2, X_3\}$ вектор узунлигини топиш учун процедурадан фойдаланинг. Унинг ҳадлари мос равишда A (4,4), B (3,3), C (2,2) матрицалар ҳадларининг йиғиндисига тенг.

2.1.15. ФАЙЛЛАР

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

- натижаларни файлга чиқарувчи ёки бошланғич маълумотларни файлдан ўқувчи дастурда **text** туридаги файлли ўзгарувчи эълон қилинган бўлиши керак;
- аниқ файлга эришиш учун файлли ўзгарувчини бу файл билан боғлаш керак (бу **assign** кўрсатмаси ёрдамида бажарилади);
- файлдан фойдаланиш мумкин бўлиши учун у очиқ бўлиши керак (**reset** – ўқиш учун, **rewrite** – ёзиш учун, **append** – қўшимча киритиш учун);
- файллар билан ишлашда хатоликлар бўлиши мумкин, масалан, дастур, бўлмаган файлни очишга уринган пайтда. Шунинг учун хатоликларга олиб келиши мумкин бўлган ҳар бир кўрсатмадан кейин, **IOResult** функциясини қўллаб, файл билан ишни тугатиш кодини текшириш керак: дастур файл билан бажариладиган иш натижасини назорат қила олиши учун, унинг матнига {SI - } директивани киритиш керак
- файлга ёзишни **write** ва **writeln** кўрсатмалар, ўқишни **read** ва **readln** кўрсатмалар бажаради, бунда бу кўрсатмаларнинг биринчи параметри сифатида файл ўзгарувчисини кўрсатиш керак;
- файл билан иш тугагандан кейин уни албатта **close** кўрсатмаси билан ёпиш керак;

- дастурда ўзгарувчисининг тури **text** деб эълон қилинган, ва шу дастур томонидан яратилган файлни матн муҳаррири ёрдамида кўриб чиқиш мумкин.

Изоҳ: масала шартларида дастурчи киритадиган сонлар йўзон шрифт билан ёзилган

МАСАЛАЛАР

- 180.** Компьютернинг алмаштириш дискида (**A:**) **numbers.txt** файлини ҳосил қилувчи ва унга фойдаланувчи томонидан киритилган 5 та бутун сонни ёзувчи дастур тузинг. **Norton Commander**га киритилган матн муҳаррири ёрдамида яратилган файлни кўринг, ҳар бир сон алоҳида сатрда жойлашганлигига ишонч ҳосил қилинг.
- 181. a: numbers.txt** файлига фойдаланувчи томонидан киритилган 5 та бутун сонни ёзувчи дастурни ёзинг. Матн муҳаррири ёрдамида файлда 10 та сон ётганлигини текширинг.
- 182. a:numbers.txt** файл таркибини экранга чиқарувчи дастурни ёзинг.
- 183. a:numbers.txt** файлида жойлашган ўрта арифметик қийматни ҳисобловчи дастурни ёзинг.
- 184.** Матн файлларини, масалан, **Turbo Pascal** бошланғич дастурлар файлларини, кўздан кечиришга (экранга файл таркибини чиқаради) имкон берадиган дастур ёзинг. Кўздан кечириладиган файл исми дастур ишлаш вақтида киритилиши керак. Тавсия қилинадиган экран кўриниши қуйида келтирилган:
- Матн файлини кўздан кечириш.
- Файлнинг тўла исмини киритинг ва **<Enter>** клавишини босинг →
- c:\tp\ pas \numb.pas**
- 185. A:** дискда жойлашган **phone.txt** файлига ўртоғингиз исми-шарифи ва телефон рақамини ёзувчи дастурни ёзинг. Агар файл дискда бўлмаса, дастур уни яратиши керак. Файлда маълумотларнинг ҳар бир элементи (исми-шарифи, телефон рақам) алоҳида сатрда бўлиши керак. Дастурнинг ишлаш вақтида тавсия этиладиган экран кўриниши қуйида келтирилган.
- Телефон маълумотномасига қўшимча
- Фамилия →
- Исм →
- Телефон →

Ахборот қўшилди.

Ишни тугатиш учун <Enter> ни босинг.

186. Телефон маълумотномасида (**a: phone.txt**) маълумотларни топишга имкон берувчи дастурни ёзинг. Дастур киши фамилиясини сўраши ва унинг телефон рақамини чиқариши керак. Агар маълумотномада бир хил фамилиялар бўлса, дастур шундай фамилияга эга ҳамма кишиларнинг рўйхатини чиқариши керак. Дастурнинг ишлаш вақтида тавсия қилинадиган экран кўриниши қуйида келтирилган.

Телефон маълумотномасидан излаш.

Фамилияни киритиш ва <Enter> ни босинг. Дастур билан ишни тугатиш учун таклифдан кейин дарров <Enter> ни босинг.

Фамилия → **Юсупов.**

Маълумотномада Юсупов ҳақида ахборот йўқ.

→ **Исаев.**

Исаев Ҳамро 578 – 12 – 45

Исаев Ҳасан 224 – 34 – 02

187. Тестдан ўтказиш универсал дастурини ёзинг. Тест саволлари кетма-кетлиги ва жавоб вариантлари матн файлида бўлиши керак. Дастур бу файл исмини уни ишга туширувчи буйруқ сатридан олиши керак. Тест саволлари сони чекланмаган. Лекин қуйидаги чекликларни киритиш таклиф этилади: савол матни ва альтернатив жавоблар экраннинг битта сатридан ортиқ жойни эгалламаслиги керак.

Дастур баҳоларни қуйидаги қоидага кўра қўйиш керак. Ҳамма жавоблар тўғри бўлса – аъло, 80% дан кам бўлмаган саволларга тўғри жавоб берилса – яхши, 60% дан кам бўлмаган тўғри жавобларга қониқарли, 60% дан кам тўғри жавоблар учун ёмон.

Қуйида тест саволлари файлининг тавсия этиладиган таснифи келтирилади (N_i - i - саволга альтернатив жавоблар сони; K_i - тўғри жавоб рақами), тест файлига мисол ва экраннинг дастур ишлаётган вақтидаги кўриниши қуйида келтирилган.

Савол ₁
N_1M_1
Жавоб
...
Жавоб
Савол ₂

N_2M_2

Жавоб

...

Жавоб

Савол_к

N_kM_k

Жавоб

...

Жавоб

div нимани билдиради?

3.2.

бўлишни

бутун бўлинмани

бўлишдан ҳосил бўлган қолдиқни

Паскал тили асосчиси ким?

2.2.

Норберт Винер

Никлас Вирт

Қуйидаги мантиқий амаллардан қайси бири биринчи бўлиб бажарилади?

3.2.

and

not

or

Ҳозир Сизга тест таклиф этилади. Ҳар бир саволга бир нечта жавоблар варианты берилади. Сиз тўғри жавоб рақамини киритишингиз ва <Enter> ни босишингиз керак.

div нимани билдиради?

1. бўлишни

2. бутун бўлинмани

3. бўлишда ҳосил бўлган қолдиқни

→ 2

Паскал тили асосчиси ким?

1. Норберт Винер
2. Никлас Вирт

→ 2

Қуйидаги мантиқий амаллардан қайси бири биринчи бўлиб бажарилади?

1. and
2. not
3. or

→ 2

Баҳонгиз «Аъло»!

Дастур ишини якунлаш учун <Enter> ни босинг.

188. Дюймлардан миллиметрларга ўтиш жадвалини, фойдаланувчи хоҳишига кўра, экранга, принтерга ёки файлга чиқариш дастурини ёзинг. Қуйида дастурнинг ишлаш вақтида тавсия қилинадиган экран кўриниши келтирилган.

*** Дюймлардан миллиметрларга утиш жадвали ***

Натижа чиқаради:

- 1 – экранга
- 2 – принтерга
- 3 – файлга

1 дан 3 гача сонларни киритинг ва <Enter> ни босинг.

Сизнинг танловингиз → 2

Дюймлар	Миллиметрлар
0.5	12.7
1.0	25.4
1.5	38.1
2.0	50.8
2.5	63.5

Дюймлар	Миллиметрлар
3.0	76.2
3.5	88.9
4.0	101.6
4.5	114.3
5.0	127.0

189. a:\phone.txt файлига бир сеансда бир нечта одамлар тўғрисида ахборот киритиш дастурини ёзинг.

190. Телефон маълумотномасига қўшимча киритадиган, маълумотномада излаш ишларини бирлаштирадиган дастур тузинг.

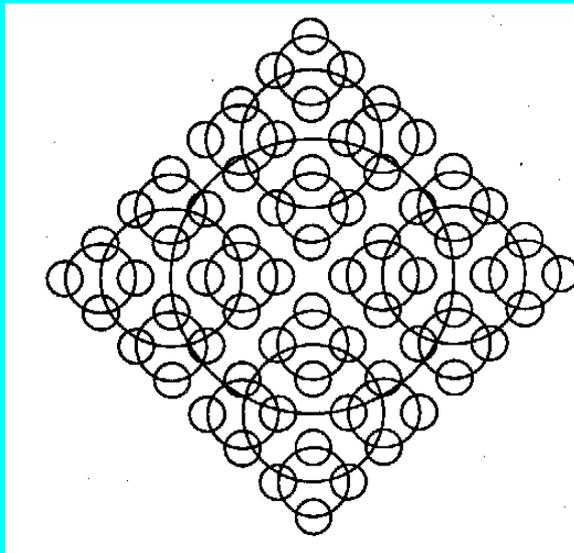
2.1.16. РЕКУРСИЯ

Бу бўлим масалаларини ечишга ўтишдан олдин қуйидагиларни эслаш керак:

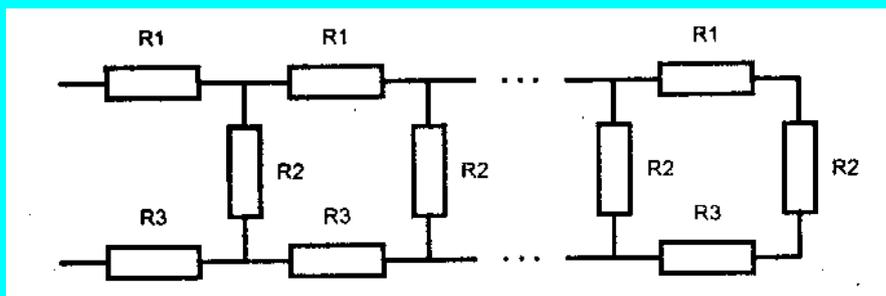
- ўз-ўзини чақириб олувчи процедура ва функция рекурсив дейилади;
- жараёни тугатиш учун рекурсив функция (процедура) алгоритмида функциянинг (процедуранинг) бевосита тугашини таъминловчи йўналиш, албатта, бўлиши керак.

191. Факториални ҳисобловчи функцияни ва унинг ишлашини текширувчи дастурни ёзинг.

192. Қуйида келтирилган безакни экранга чиқарувчи дастурни ёзинг.



193. Схемаси қуйида келтирилган электр занжир қаршилигини ҳисоблаш дастурини ёзинг. Қаршилиқлар миқдори ва занжир тартиби (R_2 қаршилиқлар сони) дастур ишлаши вақтида киритилиши керак.



194. Юқорида келтирилган электр занжир схемасини экранга чизувчи дастурни ёзинг. Занжир тартиби дастур ишлаш вақтида киритилиши керак.

195. Йиғиндини X нинг ҳақиқий қийматларида ҳисобланг. $S = \sum_{n=1}^k \frac{X^{2n}}{(2n)}$

196. Йиғиндини X нинг ҳақиқий қийматларида ҳисобланг. $S = \sum_{n=1}^k \frac{X^{4n+1}}{4n+1}$

197. Йиғиндини X нинг ҳақиқий қийматларида ҳисобланг. $S = \sum_{n=1}^k (-1)^n \frac{\cos nx}{n^2}$

198. Йиғиндини X нинг ҳақиқий қийматларида ҳисобланг. $S = \sum_{n=1}^k \frac{n^2 + 1}{n!} \left(\frac{X}{2}\right)^n$

199. $y=e^x$ функциянинг қийматини аргументнинг қуйидига ўзгаришлар соҳасидаги $1 \leq X \leq 2$ ҳар бир нуқтасида қаторда ёйиб ($n=15$) ҳисобланг.

Қатор йиғиндисининг ҳар умумий ҳади: $S = \sum_{n=1}^k \frac{X^n}{n!}$

§ 2.2. ЮҚОРИ МУРАККАБЛИКДАГИ МАСАЛАЛАР

Бу масалаларни беришдан мақсад қобилятли талабаларни дастурлаш билан шугулланишга қизиқтириш, уларга ўз устларида ишлашлари учун маълумот бериш, уларнинг кўникма, маҳорат, қобилятларини, касб масалакасини ошириш

- 200. «Кўндаланг диагонал».** $A(m,n)$ массивнинг, индекслари айирмаси берилган K сонига (K -манфий сон ҳам бўлиши мумкин) тенг бўлган ($i-j=k$), элементлари йиғиндисини топинг.
- 201. «Квадратчалар».** Ҳар бир элементи $0,1,5$ ёки 11 га тенг $A(m, m)$ массив берилган. Ҳар бирида элементлари ҳар хил бўлган тўртликлар $A(i,j)$, $A(i+1,j)$, $A(i,j+1)$, $A(i+1,j+1)$ миқдорини топинг.
- 202. «Санок системалари».** $M(9)$ массивда қандайдир натурал соннинг I – санок системасида рақамлари ёзилган. ($M(1)$ -бирлар хонаси ва ҳ.к.). Бирлар хонасидан бошлаб туриб, J -санок системасида бу сон рақамларини чоп этинг. I, J сонлар 10 дан ошмайди.
- 203. «Календар».** Кун, ой, йилни билдирувчи учта a, b, c сонлар берилган. Шу куннинг йил бошидан ҳисоблангандаги n -тартиб рақамини топинг.
- Кўрсатма: тартиб рақами 400 га бўлинадиган, шунингдек тартиб рақами 4 га бўлиниб, 100 га бўлинмайдиган йиллар кабиса йили ҳисобланади.
- 204. «Сўзнинг матнга киритилиши».** Бутун сонли $X(n)$ ва $Y(k)$ 2 та массивлар берилган. $X_{i+1}=Y_1$, $X_{i+2}=Y_2$, ..., $X_{i+k}=Y_k$ шартларни қаноатлантирувчи ва X_{i+1} , X_{i+2}, \dots, X_{i+k} , кетма-кет келувчи «к» та элементларни биринчи массивдан танлаб олиш мумкинми? Бу масалани ечувчи ва Ҳа ёки Йўқ жавобларни берувчи дастурни ёзинг.
- 205. «Ноллар серияси».** $A(n)$ бутун қийматли массив берилган. Массивнинг элементлари нолга тенг ва кетма-кет келувчи энг узун кетма-кетлиги узунлигини топинг.
- 206. «Эгар нуқта».** $A(m,n)$ сонли массив берилган. Ўз сатрида энг кичик бўлган элемент ўз устунида энг катта бўлса, у эгар нуқта дейилади. Агар массивда эгар нуқта бўлса, у ётган сатр ва устун рақамини, агар ундай нуқта бўлмаса, нол сонини чоп этинг.
- 207. «Касрни қисқартириш».** Натурал « m » ва « n » сонлар берилган. Умумий бўлувчига эга бўлмаган шундай натурал m_1 ва n_1 сонларни топиш керакки, $m_1/n_1 = m/n$ бўлсин.

208. «Массивларнинг қўшилиши». «M» ва «N» сонлар ва иккита тартибга солинган массивлар: $a_1 \leq a_2 \leq \dots \leq a_m$ ҳамда $b_1 \leq b_2 \leq \dots \leq b_n$ берилган.

Бу элементлардан тартибга солинган учинчи массивни ҳосил қилинг: $c_1 \leq c_2 \leq \dots \leq c_{m+n}$ Кўрсатма: «M» ва «N» лар катта сонлар бўлганда дастурдаги амаллар миқдорига эътибор беринг.

209. «Мода». Бутун сонли $A(n)$ массивда энг кўп учрайдиган сонни топинг. Агар бундай сонлар бир нечта бўлса, улардан биттасини аниқланг.

210. «Марказий қишлоқ». «K» та қишлоқ мавжуд. Агар i -қишлоқда тез ёрдам пункти жойлаштирилса, қакирув бўйича j -қишлоққа бориш $a_{ii} + a_{ij}$ ($1 \leq i, j \leq k$, $i \neq j$) вақтни олади.. Шундай i -қишлоқ тартиб рақамини топингки, ундан энг узоқ қишлоққа бориш учун кам вақт сарфлансин. $A(k, k)$ массив берилган. Унда ҳамма a_{ij} элементлар нолдан катта ва a_{ij} - элемент a_{ji} элементга тенг бўлмаслиги мумкин.

211. «Тартиб индекслари». $A(n)$ сонли массив берилган . $1, 2, \dots, n$ сонларнинг шундай, i_1, i_2, \dots, i_n ўрин алмаштиришни топингки, натижада $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_n}$ бўлсин.

212. «Ноллаштириш». Берилган икки ўлчовли $A(m, n)$ массивда ноли бўлган сатр ва устун элементларини ноллар билан алмаштиринг.

Шарт. Ёрдамчи бир ўлчовли массивдан фойдаланиш мумкин, лекин ёрдамчи икки ўлчовли массив ишлатиш мумкин эмас .

213. «Улгуржи харид». Пайпоқ жуфти 105 сўм, боғлами (12 жуфт) 1025 сўм, кутиси (12 боғлам) 11400 сўм туради.

Харидор сотиб олмоқчи бўлган пайпоқларнинг n жуфт сонига кўра, харидор сотиб олиши керак бўлган n_1, n_2, n_3 кути, боғлам, пайпоқлар жуфтини ҳисоблаб берувчи дастур тузинг.

Тушунтириш. 11 жуфт пайпоқ ўрнига бир боғламни харид қилиш арзонга тушади.

214. «Тўнтарилган сонлар». $A(N)$ сонли массив берилган. Массивнинг максимал узунликдаги кесмасини топиш керак. Унда биринчи сон охиригисига, иккинчи сон охиригисидан битта олдингисига ва ҳ.к. тенг бўлсин . Бу кесма узунлигини чоп этинг.

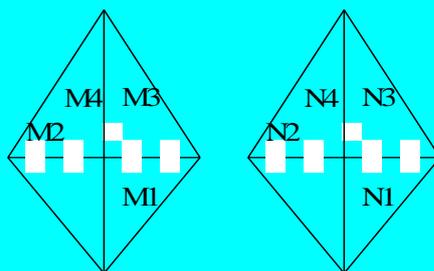
215. «Икки марта монотон». Сонларнинг $A(m, n)$ массиви сатрлар ва устунлар бўйича камайиб бориш тарзида тартибланган, яъни ҳамма $i=1, \dots, m$ лар учун, $a_{i1} \leq a_{i2} \leq \dots$ ҳамма $j=1, \dots, n$ лар учун, $a_{1j} \leq a_{2j} \leq \dots$.

Массив элементлари ичидан берилган «х» сонига тенг бўлганини топинг.

Агар бундай элемент бўлмаса, «Йўқ» деб чоп этинг. Мажбурий шарт. Ечимда амаллар сони $m \cdot n$ атрофида эмас, $m+n$ атрофида бўлсин.

216. «Тетраэдрлар». Иккита тенг тўғри M ва N тетраэдрлар қиррасига M_1, M_2, M_3, M_4 ва N_1, N_2, N_3, N_4 сонлар 3-расмда кўрсатилган тартибда ёзилган.

Тетраэдрларни бир хил сонлар ёзилган қирралари билан бир-бирига мос тушириб, жойлаштириш мумкинми? «Ҳа» ёки «Йўқ» жавоб берилсин (1-расм).



1-расм

217. «Каср даври». « M » ва « N » натурал сонлар берилган. Ўнлик M/N каср даврини чоп этинг. Масалан, $1/7$ каср учун давр 142857 га тенг, каср чекли бўлса, унинг даври битта 0 рақамига тенг бўлади.

218. «Арра». $X(m)$ массив берилган. Энг узун «арра» шаклидаги (тишлари юқорига қараган) кетма-кет келувчи сонлар қатори узунлигини топинг.

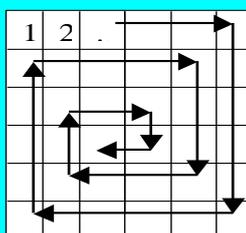
$$X[p+1] < X[p+2] > X[p+3] < \dots > X[p+k]$$

219. «Ноллар охирига». Бир ўлчовли массив берилган. Унинг нолга тенг бўлмаган ҳамма элементларини, тартибни сақлаган ҳолда, массив бошига, ноллик элементларини, янги массив ишлатмасдан туриб, массив охирига жойлаштиринг.

220. «Ҳар хил рақамлик сонлар». Ўнлик саноқ системасидаги сонлар ичидан иккита рақами бир хил бўлмаган барча тўрт хонали натурал сонларни чоп этинг.

221. «0,1,2 ларнинг ўрнини алмаштириш». $X(n)$ массивда ҳар бир элемент 0,1 ёки 2 га тенг. Массив элементлари ўрнини шундай алмаштирингки, натижада аввал ҳамма ноллар, кейин бирликлар, ниҳоят, ҳамма иккиликлар жойлашсин (қўшимча массив ишлатиш мумкин эмас).

- 222. «Арифметик амаллар».** Ёзилган $(((((1?2)?3)?4)?5)?6)$ ифодада ҳар бир ? белги ўрнига +, -, *, /, арифметик амаллардан биттасини шундай қўйиш керакки, ҳисоблаш натижаси 35 бўлсин (бўлишда бўлинмадаги каср қисм ташланади). Битта ечимни топиш етарли.
- 223. «Тез даража».** «А» ҳақиқий ва «К» натурал сонлар берилган. Даражага кўтариш амалидан фойдаланмасдан (бу ерда «К» жуда катта сон бўлиши мумкин, натижада «К» кўпайтиришни бажариб бўлмай қолади) « A^K »ни ҳисобланг ва чоп этинг.
- 224. «Ҳар хил сонлар».** «М» натурал сон ва $A(m)$ бутун қийматли массив берилган. Бу массивда нечта ҳар хил сонлар бор? Масалан: 5,7,5 дан иборат уч элементли массивда ҳар хил сонлар иккита (5 ва 7).
- 225. «Минимумларнинг максимуми».** $X(m,n)$ массивда ҳамма сонлар ҳар хил. Ҳар бир сатрдан энг кичик элементни танлаш, сўнгра бу кичик элементларнинг ичидан энг каттасини аниқлаш дастурини тузинг. X -массивда танланган сон ётган сатр ва устун рақамларини аниқланг.
- 226. «Спирал».** N сонини киритинг ва $n \times n$ ўлчовли икки ўлчамли массивни спирал бўйича 1,2,... натурал сонлари билан тўлдириг (2-расм).



2-расм

- 227. «Ички тўплам бўйича йигинди».** Бутун қийматли $A(n)$ массив ва «М» сон берилган. Элементларнинг шундай қисм тўпламини $A[i_1], A[i_2], \dots, A[i_k]$ ($1 \leq i_1 < \dots < i_k \leq n$) топиш керакки, $A[i_1] + A[i_2] + \dots + A[i_k] = M$ бўлсин. Бундай тўплам бор, деб фараз қилинади.
- 228. «Лабиринт».** Йўловчи лабиринтдан (боши берк йўл) чиқа оладими? Агар чиқа олса, унда йўловчининг чиқишдан бошланғич нуқтасигача бўлган йўлини чоп этинг. Лабиринт 40×40 ўлчамли «А»массив билан берилган. Унда агар (k,m) катакдан ўтиш мумкин бўлса, $a[k,m]=0$, (k,m) катакдан ўтиш мумкин бўлмаса, $a[k,m]=1$. Йўловчининг бошланғич нуқтаси, ўтиш мумкин бўлган, $[i,j]$ катакда. Йўловчи, агар катаклар битта умумий томонга эга бўлса, бир катакдан иккинчисига ўта олади.

Йўловчи чегара катакка (яъни (к,м), бу ерда «к» ва «м» 1 ёки 40 га тенг) етсагина лабиринтдан чиқади.

229. «Мукаммал сонлар». Агар натурал сон ўзининг барча бўлувчилари (1 ҳам ҳисобланади) йиғиндисига тенг бўлса, у мукаммал сон бўлади. Берилган М сонидан кичик барча мукаммал сонларни топиш дастурини тузиш.

230. «Ҳосил қилинмайдиган сон». $P(n)$ натурал сонлар массиви берилган. Шундай энг кичик натурал сонни топиш керакки, уни P массивнинг ҳеч бир элементлари йиғиндиси кўринишида ифодалаб бўлмасин. Йиғинди битта қўшилувчидан ҳам иборат бўлиши мумкин, лекин массивнинг ҳар бир элементи унга фақат бир марта кира олади.

231. «Дўмбира». Айлана бўйича 12 та сон ёзилган : a_1, a_2, \dots, a_{12} . Агар k -тартиб рақамидан бошлаб, уларни қўшиб ёзсак, X_k вектор ҳосил бўлади:

$$X_k = (a_k, a_{k+1}, a_{k+11}),$$

Бу ерда a_{13} ни a_1 , a_{14} ни a_2 ва ҳ.к. билдиради. Агар биринчи тенгмас жуфтдаёқ $a_{k+i} < a_{p+i}$ ($i=0,1,\dots$) бўлса, X_k вектор X_p вектордан кичик деб ҳисобланади. Шундай «к» ни топиш керакки, вектор X_k энг кичик бўлсин.

232. «Кублар йиғиндиси». Берилган N натурал сонни 2та натурал сонлар кубларининг йиғиндиси ($N=i^3+j^3$) кўринишида неча хил усул билан ёзиш мумкин?

Қўшилувчиларнинг ўрин алмаштирилиши янги усулни бермайди.

1/3-даражага кўтариш амалидан фойдаланиш мумкин эмас.

233. «Оддий бўлувчилар». Натурал N сон берилган . Унинг барча оддий бўлувчиларини топинг.

234. «Кўпҳад». Берилган ҳақиқий X_1, X_2, \dots, X_n илдизларга эга $P(x)=a_0+a_1x+a_2x^2+\dots+a_{n-1}x^{n-1}+x^n$ кўпҳад берилган. Унинг a_0, a_1, \dots, a_{n-1} коэффициентларини топинг.

Эслатма: Безу теоремасига кўра $P(x) = (x-x_1)*(x-x_2)*\dots*(x-x_n)$

235. «Қисқа кўпайтувчилар» A сонлар тўплами қуйидаги шартлар билан берилган:

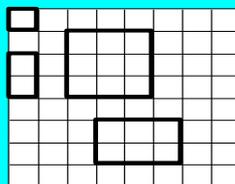
а) $1 \in A$

б) агар $k \in A$ бўлса , $2*k+1 \in A$ ва $3*k+1 \in A$ бўлади ва A тўплам бошқа сонларга эга бўлмайди. A тўпламнинг дастлабки $n < 1000$ сонларини ўсиб бориш тартибида чоп этинг. Чоп этиладиган сонларнинг бошланиши қуйидагича: 1,3,4,7,9,10,13,15,19,...

- 236. «Покер».** Бешта сондан иборат массив берилган. Уларнинг ичидан бештаси бир хил бўлса-1; тўрттаси бир хил бўлса-2; учтаси ва иккитаси бир хил бўлса-3; учтаси бир хил бўлса-4; иккита иккиталик бир хил бўлса-5; иккитаси бир хил бўлса-2; акс ҳолда-7 сонини чоп этинг.
- 237. «Тенг элементлар».** Бутун сонли $A(m,n)$ массив берилган. Массивнинг ҳар бир сатри камайиб бориш тарзида тартибга солинган, яъни: ҳамма i лар учун $i=1, \dots, m$; $a_{i1} \leq a_{i2} \leq \dots$ ҳамма сатрларда учрайдиган сонни топинг ва чоп этинг. Агар бундай сон бўлмаса, «йўк» деган жавоб ҳосил қилинг.
- 238. «Йўл».** m ва $n \geq 2$ натурал сонлар ҳамда ҳақиқий $A(m,m,n-1)$ массив берилган. Бутун сонларнинг мумкин бўлган ҳамма $1 \leq i_1, i_2, \dots, i_n \leq m$ тўпламлари учун қуйидаги йиғиндининг минимал қийматини топинг:

$$R = A(i_1, i_2, 1) + A(i_2, i_3, 2) + A(i_{n-1}, i_n, n-1)$$
Изоҳ: ва n сонлар бир неча ўнталикка тенг миқдорлар. Шунинг учун m^n тартибли амаллар сонигача ечиш ноўрин.
- 239. «Учбурчак ва нуқта».** Учбурчак учларининг тўғри бурчак координаталари X_1, Y_1 ; X_2, Y_2 ; X_3, Y_3 ва нуқтанинг X, Y координаталари берилган. Нуқта учбурчакда ётиш-ётмаслигини аниқланг. Ҳисоблаш хатоликларини эътибордан четда қолдиринг.
- 240. «Тартибланган касрлар».** Махражи 7 дан катта бўлмаган қиймати 0 ва 1 ўртасида бўлган қисқармайдиган ҳамма оддий касрларни ўсиб бориш тартибида чоп этинг.
- 241. «Тўртликлар жуфти».** Ягона ҳол бўлмаган тўртта натурал сонлар квадратлари йиғиндисини ифодаловчи энг кичик сонни топинг.
- 242. « $M+1$ сонни иккиликда ёзиш».** Бутун манфиймас « M » сон ўзининг иккилик a_0, a_1, \dots, a_{n-1} рақамлари билан берилган $M = a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \dots + a_1 * 2 + a_0$, бу ерда a_i нолга ёки бирга тенг ($i=0, 1, \dots, n-1$). $M+1$ сонининг иккилик рақамлар массивини ёзинг.
- 243. «Квадратлар йиғиндисини».** Берилган M натурал сонни иккита натурал сонлар квадратлари йиғиндисини кўринишида ёзиш мумкинми?
- 244. «Ўрин алмаштириш».** Ҳар хил сонлардан иборат « M » та жуфт сонларнинг $A(M)$ массиви берилган. Сонларнинг барча ўрин алмаштиришларини ҳосил қилинг.
- 245. «Туб сонлар».** Берилган M сонидан катта бўлмаган барча туб сонларни босмадан чиқаринг.

- 246. «Тенг сонларни излаш».** Бутун қийматли 2 ўлчамли $A(2,15)$ массив берилган. Унинг элементларидан иккитаси ва фақат иккитасининг ўзаро тенглиги маълум. Уларнинг индексини аниқланг.
- 247. «Рақамларнинг берилган йиғиндиси».** Рақамлари йиғиндиси берилган натурал сонга тенг бўлган барча уч хоналик ўнлик санок системасидаги сонларни аниқланг.
- 248. «Функция».** Бутун манфиймас «N»лар учун функция қуйидагича аниқланган: $f(0)=0$, $f(1)=1$, $f(2n)=f(n)$, $f(2n+1)=f(n)+f(n+1)$). Берилган N учун $f(N)$ ни топинг. Зарурий шарт: N шунчалик катта сонки, N та сондан массив ҳосил қилиш мумкин эмас (M сонинг ўсиши билан массив ҳам ўсаверади).
- 249. «Тўртбурчаклар».** 100×100 катакчали ўлчамдаги қоғоз вараққа бир нечта тўртбурчаклар чизилган. Ҳар бир тўртбурчак бутун катаклардан иборат, ҳар хил тўртбурчаклар бир-бирининг устига ётмайди ва бир-бирига урилмайди. (3-расм), 100×100 ўлчовли массив берилган, унда агар $[i,j]$ катак қандайдир бир тўртбурчакка тегишли бўлса, $A[i,j]=1$ ва акс ҳолда $A[i,j]=0$. Тўртбурчаклар сонини ҳисоблаб, чоп этувчи дастурни ёзинг.



3-расм.

- 250. «Бит – реверс».** Бутун мусбат «M» сони иккилик санок системасида ёзилади ва бу ёзувда хоналар тескари тартибда, ўнлик санок системасида қўйилади. Ҳосил бўлган сон, $V(M)$ функция қиймати, деб қабул қилинади. $M=512, 513, 514, \dots, 1023$ учун $V(M)$ функция қийматини чоп этинг. Аниқлик учун чоп этиладиган қийматлар боши: 1, 513, 257, ...
- 251. «Инверсия».** $1, 2, \dots, n$ сонларининг ўрин алмаштириши $P=(P_1, \dots, P_n)$ бўлсин. P ўрин алмаштиришнинг инверсия жадвали шундай $T=(t_1, \dots, t_n)$ кетма-кетликка тенг бўладики, унда « t_i », P ўрин алмаштиришнинг i сонидан чапдаги « i » дан катта бўлган элементлар сонига тенг бўлади. Масалан: $1, 2, \dots, 9$ сонларининг ўрин алмаштириши $P=(5, 9, 1, 8, 2, 6, 4, 7, 3)$ бўлсин, унинг инверсия жадвали қуйидагича ҳосил қилинади: $i=1$ чапда

бу сондан катта 2 та сон (5,9) бор, демак, $t_1=2$; $i=2$, чапда бу сондан катта 3 та сон (5,9,8) бор, демак, $t_2=3$ ва шундай давом эттирилади. Натижада инверсия жадвали куйидагига тенг бўлади: $T=(2,3,6,4,0,2,2,1,0)$. Берилган инверсия жадвалига кўра сонлар ўрин алмаштиришини тиклаш дастурини тузинг.

252. «Кўшилувчиларга ажратилиш». Берилган N натурал сонни кўшилувчиларга ажратилган ҳамма кўринишларини ҳосил қилинг. Кўшилувчиларнинг ўрин алмаштирилиши янги кўриниш деб ҳисобланмайди.

253. «Ўнг катта». Мусбат сонларнинг $A(N)$ массиви берилган. Ҳар бир a_i учун a_j элементни танлаймиз, a_j элемент куйидагича танланади: у a_i дан кейин келиши, a_i дан катта ва j -тартиб рақами бўйича энг кичик бўлиши керак. a_i қиймат a_j элементи билан алмаштирилади. Агар шундай a_j элемент бўлмаса, a_i қиймат нол билан алмаштирилади. Ҳосил бўлган массив чоп этилсин.

Зарурий шарт: Ечимларда амаллар сони $n \cdot n$ эмас, n та бўлиши керак. Ёрдамчи массивлардан фойдаланиш мумкин.

Тушунтириш: масалан, 2,9,8,5,9,3,4,5,2, массив алмаштиришдан кейин 9,0,9,9,0,4,5,0,0, кўринишда бўлади.

254. «Рюкзак». Берилган n предметдан шундайларини танлаб олиш керакки, уларнинг жами оғирлиги 30 кг дан кичик, қийматлари эса энг катта бўлсин. Танланган предметларнинг жами қиймати чоп этилсин.

Аниқроғи – 2та мусбат сонли $A(n)$ ва $B(n)$ массивлар берилган. Шундай ҳар хил жуфтли i_1, i_2, \dots, i_k сонларни танлаш керакки, натижада

$$a_{i_1} + a_{i_2} + \dots + a_{i_k} < 30$$

$$b_{i_1} + b_{i_2} + \dots + b_{i_k} = \max$$

бўлсин, фақат мах миқдорини чоп этинг.

Изоҳ. Предметларни, a_i –оғирлик, b_i –қиймат, b_i/a_i - баҳо ёки яна қандайдир бир бошқа белгига кўра ўсиб бориш ёки камайиб бориш тартибида жойлаштирилган, деб ҳисоблаш мумкин.

III ҚИСМ. МАСАЛАЛАР ДАСТУРЛАРИ

§ 3.1. ҚҰЙИ МУРАККАБЛИҚДАГИ МАСАЛАЛАР ДАСТУРЛАРИ

3.1.1. ЧИҚАРИШ КЎРСАТМАЛИ ДАСТУРЛАР

20 – масала

begin

```
writeln ('Инсон ақли-галактика йўлида чироқ, ');
writeln ('Балки космос мозорида кўмилар танним? ');
writeln ('Йўқ – йўқ, инсон даричасин очган у чоқда,');
writeln ('Тарихларга-у, ўлмас, деб ёзди, Ватаним. ');
writeln;
writeln (' Т.Ҳамид ');
readln; {Экрандан шеър йўқолмаслиги учун}
```

end.

25 – масала

uses Crt;

begin

```
TextBackGround (Blue);    {таг ранги}
TextColor (LigthGray);    {белгилар ранги}
ClrScr;                    {экранны тозалаш}
writeln (' Дунё бамисоли бир заррин кўза, ');
writeln (' Суви гоҳ аччиғу, гоҳ ширин бўза, ');
writeln (' Эй, инсон, умрингга бино қўйма кўп, ');
writeln (' Қазойи – муаллақ турар бош узра ');
writeln ;
writeln (' Паҳлавон Маҳмуд ');
readln; {Экрандан шеър йўқолмаслиги учун}
```

end.

26 - масала

Uses Crt;

begin

```
TextBackGround (Black);
```

```

ClrScr;
TextColor (Red);
write (' Қизил ');
TextColor (LighRed); {олов рангни очик қизил билан алмаштирамиз}
write ('Очик қизил');
TextColor (Yellow);
write ('сарик');
TextColor (Green);
write (' яшил ');
TextColor (LighBlue);
write (' ҳаво ранг ');
TextColor (Blue);
write (' кўк ');
TextColor (Magenta);
write (' бинафша ранг ');
readln; {ифода экрандан йўқолмаслиги учун}
end.

```

3.1.2. ЧИЗИҚЛИ ТАСНИФЛИ ДАСТУРЛАР

34 – масала

(Икки томони ва улар орасидаги бурчагига кўра учбурчак юзасини топиш)

var

a,b: real; {томонлар узунлиги}

f: real; {градусларда ифодаловчи бурчак катталиги}

s: real; {учбурчак юзи}

begin

writeln ('Учбурчак юзини топиш');

writeln (' Учбурчак томонлари узунлиги (см):')

write (' - > ');

readln (a, b);

writeln ('Учбурчак икки томони ўртасидаги бурчак катталиги (градус): ');

write (' - > ');

readln [f];

{ $s = a * h/2$, h (учбурчак баландлиги) $h=b*\sin(f)$ формула билан ҳисобланиши мумкин. Лекин, Turbo Pascalда Sin функциянинг аргументи радианларда ифодаланиши керак (1 рад. = $180/3.1415925$, бу ерда 3.1415926 - "ПИ" сони)}

```
s:=a*b*sin (f*3.1415926/180) /2;  
writeln;  
writeln ('учбурчак юзи:',s:6:2, 'см.кв.');
```

end.

35 - масала

(Иккита параллел уланган элементлардан иборат электр занжир қаршилигини ҳисоблаш)

var

```
r1,r2: real; {занжир элементлари қаршилиги};  
ч: real; {занжирнинг умумий қаршилиги}
```

begin

```
writeln ('Элементлари параллел уланган электр занжир қаршилигини  
ҳисоблаш')  
writeln ('Бошланғич қийматларни киритинг:');  
write ('Биринчи қаршилик катталиги (Ом) —>');  
readln (r1) ,-  
write('Иккинчи қаршилик катталиги (Ом) —>');  
readln (r2);  
r:= r1 * r2 / (r1 + r2);  
writeln;  
writeln ('Занжир қаршилиги:',ч:6:2, ' Ом');
```

end.

36 - масала

(Дала ҳовлига бориш ва қайтиш қийматларини ҳисоблаш)

var

```
mas: real; {дала ҳовлигача масофа}  
sarf: real; {100 км га бензин сарфи}
```

```
baho: real; {1 литр бензин баҳоси}
summ: real; {дала ҳовлига бориш ва қайтиш қиймати}
```

begin

```
writeln ('Дала ҳовлига бориш ва қайтиш қийматини ҳисоблаш');
write (' Дала ҳовлигача масофа (км) → ');
readln (mas);
write (' Бензин сарфи (100 км масофага, литр) → ');
readln (sarf);
write (' Литр бензин баҳоси (сўм) → ');
readln (baho);
summ: = 2 * mas / 100 * sarf * baho;
writeln;
writeln ('Дала ҳовлига бориш ва қайтиш қиймати', summ:б:2,' сўм.');
```

```
readln;
```

end.

37 – масала

{Цилиндр сирт юзасини ҳисоблаш}

var

```
r: real; {цилиндр асоси радиуси}
h: real; {цилиндр баландлиги}
s: real; {цилиндр сирт юзаси}
```

begin

```
writeln ('Цилиндр сирт юзасини ҳисоблаш');
writeln ('Бошланғич қийматлар:');
write (' Асос радиуси (см) → ')»'
readln (r) ;
write ('Цилиндр баландлиги (см) → ');
readln (h);
```

{ π — номланган ўзгармас, унинг қиймати "ПИ" сонига тенг. Sqr — Turbo Pascal функцияси, унинг қиймати аргумент квадратига тенг, яъни $Sqr(x) = x * x$ }

```
s: = 2 * pi * sqr(r) + 2 * pi * r * h;
writeln;
writeln ('Цилиндр сирт юзаси', s:б:2,'см.кв.');
```

```

readln;
end.

```

41 - масала

(Верстлардаги масофани километрларда ҳисоблаш)

```

var
v: real; {верстлардаги масофа}
k: real; {километрлардаги масофа}
begin
writeln (' Масофани Верстлардан километрларга айлантириш);
writeln ('Верстлардаги масофани киритинг ва <Enter>ни босинг');
write ('→ ');
readln (v) ,-
k: = v * 1.0668;
writeln (v:6:2,' верст – бу ', k:6:2,' км');
writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
readln;
end.

```

3.1.3. IF ТАНЛОВ КЎРСАТМАЛИ ДАСТУРЛАР

43 – масала

{Омонатларга кўра даромадни ҳисоблаш }

```

var
sum: real; {жами омонат}
muddat: real; {омонат муддати}
foiz: real; (кўшимча фоиз}
daromad: real; {омонатга кўра даромад}
begin
writeln ('Омонатга кўра даромадни ҳисоблаш. ');
writeln ('Бошланғич қийматларни киритинг: ');
write ('Омонат миқдори (сўм.) → ');
readln (sum);
write ('Омонат муддати (кунларда) → ');
readln (muddat);
write ('Кўшимча фоиз (йиллик) → ');

```

```

readln (foiz);
daromad: = (sum * foiz / 100) / 365 * muddat;
           {365 – бир йилдаги кунлар сони}
sum: = sum + daromad;
writeln;
writeln ('-----');
writeln ('Даромад:', daromad:9:2,' сўм. ');
writeln ('Омонат муддати охиридаги йиғинди:',sum:9:2, 'сўм. ');
readln;
end.

```

44 – масала

{Минутларда берилган катталиқни унга мос келувчи соат ва минутларда ифодаланган қийматга айлантириш}

var

```

min: integer; {минутларда ифодаланган интервал}
t: integer; {соат миқдори}
m: integer; {минут миқдори}

```

begin

```

writeln (' Вақт интервали миқдорини киритинг (минутларда) ва Enterни
        босинг');
write ('→ ');
readln (min);
t:= min div 60;
m:= min mod 60;
writeln;
writeln (min, ' мин. – бу ', t,' соат.',m,' мин. ');
readln;

```

end.

45 – масала

{Сонни пул форматида ифодалаш}

var

```

n: real; {каср сон}
r: integer; {соннинг бутун қисми (сўм)}

```

k: integer; {соннинг каср қисми (тийин)}

begin

writeln ('Сонни пул форматига ўзгартириш.').

write ('Каср сонини киритинг → ');

readln (n);

r:= Round (n*100) **div** 100;

k:= Round (n*100) **mod** 100;

writeln;

writeln (n:6:2,' сўм. - бу ',r,' сўм.',k,' тийин.');

readln;

end.

46 – масала

{Бўлинмани ҳисоблаш}

var

a,b,c: real; {бўлинувчи, бўлувчи ва бўлинма}

begin

writeln ('Бўлинмани ҳисоблаш);

writeln ('Битта сатрдан бўлинувчи ва бўлувчини киритинг,');

writeln ('кейин <Enter>ни босинг');

write ('→');

readln (a, b);

if b <> 0 **then**

begin

c:=a/b;

writeln (a ' a:6:3'ни ' b:6:3 ' га бўлишдан олинган бўлинма: 'c:6:3 'га
тенг);

end

else

writeln ('Хато! Бўлувчи нолга тенг бўлмаслиги керак!');

readln;

end.

47 - масала

{Ҳалқа юзасини ҳисоблаш}

```

var
    rl,r2 : real; {ҳалқа ва тешиқ радиуси }
    s : real;     {ҳалқа юзаси}
begin
    writeln ('Ҳалқа юзини ҳисоблаш:');
    writeln ('Бошланғич қийматларни киритинг:');
    write ('ҳалқа радиуси (см) →');
    readln (r1);
    write ('тешиқ радиуси (см) → ');
    readln (r2);
    if r1 > r2 then
        begin
            s:=2*3.14 * (r1-r2);
            writeln ('Ҳалқа юзаси ', s:6:2,' кв.см');
        end
    else writeln ('Хато! Тешиқ радиуси ҳалқа радиусидан катта
        бўлмаслиги керак');
        writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
        readln;
end.

```

48 - масала

{Электр занжир қаршилигини ҳисоблаш}

```

var
    r1,r2; real; {занжир қаршилиги катталиги}
    n: real; {йиғинди қаршилик}
    t: integer; {элементларни улаш тури:
        1 – кетма-кет;
        2 – параллел}

```

```

begin
    writeln ('Электр занжир қаршилигини ҳисоблаш');
    writeln ('Бошланғич қийматларни киритинг:');
    write ('Биринчи қаршилик миқдори (Ом) → ');
    readln (r1);
    write ('Иккинчи қаршилик миқдори (Ом) → ');

```

```

readln (r2);
write ('Элементларни улаш тури (1-кетма-кет, 2-параллел) → ');
readln (t);
if t=1 then r:= r1+r2
else r:=r1*r2/(r1+r2);
writeln ('Занжир қаршилиги: ',r:5:2, ' Ом');
readln;
end

```

49 - масала

```

{Квадрат тенгламани ечиш}
var
a,b,c:real; {тенглама коэффициентлари}
x1,x2:real; {тенглама илдизлари}
d:real; {дискриминант}
begin
writeln ('*Квадрат тенгламани ечиш *');
write ('Битта сатрдан коэффициентлар қийматини киритинг');
writeln(' ва <Enter>ни босинг');
write('→ ');
readln (a,b,c); {коэффициентларни киритиш}
d:= b*b - 4*a*c; {дискриминантни ҳисоблаш}
if d >= 0
then
begin
x1:= -b+sqrt(d)/(2*a);
x2:= -b-sqrt(d)/(2*a);
writeln ('Тенглама илдизлари:');
writeln ('x1 =',x1:9:3);
writeln ('x2=',x2:9:3);
end
else
writeln ('Ҳақиқий илдизлар йўқ');
end.

```

50 - масала

```
{Имтиёзни ҳисобга олган ҳолда харид қийматини ҳисоблаш}
var
summ: real; {харид жаами}
begin
writeln ('Имтиёзни ҳисобга олган ҳолда харид қийматини ҳисоблаш');
writeln ('Харид қийматини киритинг ва <Enter>ни босинг. ');
write ('→');
readln (summ);
if summ > 5000
then {имтиёз берилади }
begin
if summ > 10000
then
begin
writeln ('Сизга 5 % имтиёз берилади');
summ:=0.95*sum;
end
else
begin
writeln ('3 % имтиёз берилади');
summ: = 0.97 * summ;
end;
writeln ('имтиёзни ҳисобга олган ҳолда харид жаами:
summ:6:2, ' сўм. ');
end
else
writeln ('Имтиёз берилмайди');
readln;
end.
```

52 - масала

```
{Тарихни билишни текшириш}
var
jav: integer; {синалувчининг жавоби}
```

```

begin
  writeln (' А.Навоий қачон туғилган?')
  writeln ('Сонни киритинг ва <Enter>ни босинг');
  write('→');
  readln (jav);
  if jav = 1441
  then writeln ('Тўғри.')
  else
    begin
      write {'Адашдингиз, '};
      writeln ('А.Навоий 1441 йилда туғилган,')
    end;
  readln;
end.

```

53 - масала

```

{Фанлар тарихидан билимни текшириш}
var
  jav: integer; {синалувчи танлаган жавоб рақами}
begin
  writeln ( ' Кибернетика фани асосчиси ким?');
  writeln ('1. Андре Мари Ампер');
  writeln ('2 . Исаак Ньютон ' ) ;
  writeln('3. Норберт Винер');
  writeln;
  writeln ('тўғри жавоб рақамини киритинг ва <Enter> ни босинг') ;
  write ('→' );
  readln (jav) ;
  if jav = 2
  then writeln ('Тўғри.')
  else
    begin
      writeln ( ' Адашдингиз. ' ) ;
      writeln ('Кибернетика фани асосчиси - Норберт Винер. ');
    end;

```



```

if p = jav
    then writeln ('Тўғри.')
    else writeln ('Сиз адашдингиз. ',m1,'x',m2,',p);
readln;
end.

```

59 - масала

```

{Клавиатурадан киритилган соннинг жуфтлигини текшириш.}
var
n: integer; {фойдаланувчи киритган сон}
begin
writeln ('Бутун сонни киритинг ва <Enter>ни босинг. ');
write ('→ ');
readln (n);
write ('Сон', n, ' - ');
if n mod 2 = 0
    then writeln (n, ' сони жуфт.')
    else writeln (n, ' сони тоқ. ');
readln; {натижа экрандан йўқолмаслиги учун}
end.

```

61 - масала

{Шанба ва якшанба кунлари бериладиган имтиёзни ҳисобга олган ҳолда телефондан сўзлашув қийматини ҳисоблаш }

```

var
Time:integer; {сўзлашув узунлиги}
Day:integer; {ҳафта куни}
Summa:real; {сўзлашув қиймати}
begin
writeln ('Телефонда сўзлашув қийматини ҳисоблаш. ');
writeln ('Бошланғич қийматларни киритинг:1);
write ('Сўзлашув узунлиги');
write (' (минутларнинг бутун миқдори) —>');
readln (Time);
write ('Ҳафта куни');

```

```

write (' (1-душанба, ... , 7-якшанба) → ');
readln (Day);
Summa:= 20.3 * Time;           { 1 минут баҳоси 20.3 сўм. }
if (Day = 6) or (Day = 7)
then begin
    writeln ('20 % ли имтиёз берилади. ');
    Summa:=Summa * 0.8;
end,
writeln ('Сўзлашув қиймати:',Summa:8:2,' сўм. ');
readln;
end.

```

62 - масала

```

{Вазни назорат этиш}
var
    w:real; {вазн}
    h:real; {бўй}
    opt: real; {оптимал вазн}
    d:real; { оптимал вазндан фарқланиш }
begin
    writeln ('Бир сатрдан бўшлиқ ташлаб киритинг');
    writerln ('бўй (см) ва вазн (кг), кейин <Enter>ни босинг');
    write ('→');
    readln (h,w) ;
    opt:=h-100;
    if w = opt
    then
        writeln ('Сизнинг вазнингиз оптимал!')
    else
        if w<opt
        then begin
            d:=opt-w;
            writeln ('Сизга', d:5:2, ' кг.га семириш керак');
        end
    else

```

```

begin
    d:=w-opt;
    writeln ('Сиз', d:5:2, ' кг.га озишингиз керак');
end;
readln;
end.

```

3.1.4. CASE КҮРСАТМАЛИ ДАСТУРЛАР

64 - масала

```

{Ой рақамига кўра йил мавсумини аниқлаш}
var
    oy: integer; {ой рақами }
begin
    writeln ('Ой рақамини киритинг (1 дан 12гача бўлган сонлар) ва <Enter>
        ни босинг');
    write ('→ ');
    readln (oy);
    case oy of
        1,2,12:    writeln ('Қиш');
        3 .. 5:    writeln ('Баҳор') ;
        6 .. 8:    writeln ('Ёз');
        9 .. 11:   writeln ('Куз');
        else       writeln ('Сон 1дан 12гача бўлиши керак');
    end;
    readln;
end.

```

65 - масала

```

{Шаҳарлараро телефонда сўзлашув қийматини ҳисоблаш}
var
    kod:    integer; {шаҳар коди }
    baho:   real; {минут баҳоси}
    uzun:   integer; {сўзлашув узунлиги}
    summ:   real; {сўзлашув қиймати}
begin

```

```

writeln ('Телефонда сўзлашув қийматини ҳисоблаш. ');
writeln ('Бошланғич қийматларни киритинг:');
write ('Шаҳар коди → ');
readln (kod);
write ('Сўзлашув узунлиги (минутнинг бутун миқдори) —>');
readln (uzun);
write ('Шаҳар: ');
case kod of
  371: begin
        writeln ('Тошкент');
        baho: = 150;
      end;
  095: begin
        writeln('Самарқанд');
        baho:=120;
      end;
  375: begin
        writeln('Қарши');
        baho:=100;
      end;
  365: begin
        writeln ('Бухоро');
        baho: = 130;
      end;
end;
summ := baho * uzun;
writeln ('Минут баҳоси:',baho:6:2,' сўм. ');
writeln( 'Сўзлашув қиймати: ',summ: 6:2, ' сўм. ');
readln;
end.

```

66 - масала

{Навбатдаги кун санасини аниқлаш }

var

kun: integer;

```
oy: integer;
yil: integer;
oxir: boolean; { TRUE, агар жорий кун ойнинг охирги куни бўлса }
r: integer;     {агар йил кабиса йили бўлса, унда улни 4 га бўлишдаги
                қолдиқ нолга тенг }
```

```
begin
```

```
write ('Бугунги санани рақамларда киритинг');
```

```
write (' (кун ой йил) → ');
```

```
readln (kun, oy, yil);
```

```
oxir: = FALSE;
```

```
case oy of
```

```
1, 3, 5, 7, 8, 10, 12 : if kun = 31
```

```
    then oxir: = TRUE;
```

```
4, 6,9,11 : if kun= 30
```

```
    then oxir: = TRUE;
```

```
2 : if kun= 28 then
```

```
    begin
```

```
        r:=yil mod 4;
```

```
        if r <> 0
```

```
            then oxir: = TRUE;
```

```
    end;
```

```
end; { case }
```

```
if oxir
```

```
    then begin
```

```
        writeln ('Ойнинг охирги куни!');
```

```
        kun:=1;
```

```
        if oy = 12
```

```
            then begin
```

```
                oy:=1;
```

```
                yil:=yil + 1;
```

```
                writeln ('Янги йилингиз билан!');
```

```
            end
```

```
        else oy: = oy + 1;
```

```
    end
```

```
    else kun:=kun + 1;
```

```
writeln ('Эрта ' ,kun,' . ', oy, ' . ',yil);
readln;
end.
```

3.1.5. FOR КЎРСАТМАЛИ ДАСТУРЛАР

68 - масала

{Биринчи ўнта бутун мусбат сонлар квадратлари жадвалини чиқариш}

var

```
x: integer; {сон}
y: integer; {сон квадрати}
i: integer; {цикл ҳисобчиси}
```

begin

```
writeln ('Квадратлар жадвали');
writeln ('-----');
writeln ('Сон  Квадрат' );
writeln (' -----') ;
```

for i:= 1 to 10 do

begin

```
y:=x*x;
writeln (x:3,y:10);
x:=x+1;
```

end;

```
writeln ('-----') ;
readln;
```

end.

69 - масала

{Биринчи n та бутун мусбат сонлар йиғиндисини ҳисоблаш}

var

```
n: integer;      {кўшиладиган сонлар миқдори}
summ: integer;  {йиғинди}
i: integer;     {цикл ҳисобчиси}
```

begin

```
writeln ('Мусбат сонлар йиғиндисини ҳисоблаш');
```

```

write ('Қўшиладиган сонлар миқдорини киритинг —>');
readln (n)
summ: =0;
for i:=1 to n do
    summ: = summ + i;
write ('Биринчи ',n, 'бутун мусбат сонлар йиғиндиси ');
writeln ('тенг' , summ) ;
readln;
end.

```

72 - масала

{1,3,5,7 ... қатор хусусий йиғиндисини ҳисоблаш }

var

```

e: integer;      { қатор ҳади }
n: integer;      { қўшиладиган ҳадлар миқдори }
summ: integer;   { қатор ҳадлари йиғиндиси }
i: integer;      { цикл ҳисобчиси }

```

begin

```

writeln ('1,3,5,7, . . қатор хусусий йиғиндисини ҳисоблаш ');
writeln ('қўшиладиган ҳадлар миқдорини киритинг —>');
readln (n);
e:=1; { қатор биринчи ҳади }
summ: = 0;
for i:=1 to n do

```

begin

```

    summ: = sum + e;      { йиғиндига қатор навбатдаги ҳадини қўшинг }
    e: = e + 2;          { кейинги ҳад қийматини ҳисобланг }

```

end;

```

writeln ('Қаторнинг биринчи ',n,' ҳадлар йиғиндиси тенг:', summ);
readln;

```

end.

73 – масала

{1 + 1/2 + 1/3 - ... қатор ҳади йиғиндисини ҳисоблаш }

var

```
n: integer; { қаторнинг қўшиладиган ҳадлар миқдори }
i: integer; { қатор элементи тартиб рақами }
elem: real; { қатор элементи қиймати }
summ: real; { қатор ҳадлари йиғиндиси }
```

```
begin
```

```
writeln ('1+1/2+1/3 + ... қатор хусусий йиғиндисини ҳисоблаш ');
writeln ('Қаторнинг қўшиладиган ҳадлар миқдорини киритинг');
write ('→');
```

```
readln (n);
```

```
summ:=0;
```

```
for i:= 1 to n do
```

```
begin
```

```
    elem:= 1/i;
```

```
    summ: = summ + elem;
```

```
end;
```

```
write('Биринчи ', n,'ҳад йиғиндиси:');
```

```
writeln (summ:6:4);
```

```
readln;
```

```
end.
```

74 - масала

```
{Иккиннинг даражалар жадвали}
```

```
var
```

```
n: integer; { даража кўрсаткичи }
```

```
x: integer; { n даражадаги 2 қиймати }
```

```
begin
```

```
writeln ('Иккиннинг даражалар жадвали');
```

```
x: = 1;
```

```
for n:=0 to 10 do
```

```
begin
```

```
    writeln (n:2, x:6);
```

```
    x:=x*2;
```

```
end;
```

```
readln;
```

```
end.
```

79 - масала

{Функциянинг жадвали}

const

B = -2; {аргумент ўзгариш соҳасининг қуйи чегараси }

A = 2; {аргумент ўзгариш соҳасининг юқори чегараси }

Dx = 0.5; { аргумент орттирмаси }

var

x, y: real; {аргумент ва функция қиймати }

n: integer; {нуқталар сони }

i: integer; {цикл ҳисобчиси }

begin

n:=Round((A - B)/DX) + 1;

x:=B;

writeln ('-----');

writeln (' x | y');

writeln ('-----');

for i:=1 **to** n **do**

begin

y:= -2.4*x*x+5*x-3;

writeln (x:8:2, '|', y:8:2);

x:=x+DX;

end;

writeln ('-----');

readln;

end.

82 - масала

{Клавиатурадан киритиладиган каср сонлар кетма-кетлигининг ўрта арифметигини ҳисоблаш }

const

L=5; {кетма-кетлик узунлиги }

var

a: real; {сон }

n: integer; {киритилган сонлар миқдори }

sum: real; {киритилган сонлар йиғиндиси }

```
o'rta: real; {киритилган сонларнинг ўрта арифметиги}
```

```
begin
```

```
writeln ('Қаср сонлар кетма-кетлигини қайта ишлаш');  
writeln ( 'Ҳар бир сон киритилгандан кейин <Enter>ни босинг')
```

```
sum:=0;
```

```
for n:= 1 to L do
```

```
  begin
```

```
    write ('→') ;
```

```
    readln (a) ;
```

```
    sum:=sum + a;
```

```
    o'rta:=sum/n;
```

```
    writeln ('киритилган сон' n:3, Йиғинди: ', sum: 6: 2);
```

```
    writeln ('Ўрта арифметик:', o'rta:6:2);
```

```
  end;
```

```
writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
```

```
readln;
```

```
end.
```

83 - масала

{Ўрта арифметикни ҳисоблаш ва клавиатурадан киритиладиган қаср сонлар кетма-кетлигидан энг кичик ва энг катта сонларни аниқлаш }

```
var
```

```
a: real;      { навбатдаги сон }
```

```
n; integer;  { сонлар миқдори }
```

```
sum: real;   { киритилган сонлар йиғиндиси }
```

```
o'rta: real; { ўрта арифметик }
```

```
min: real,-  { кетма-кетликнинг энг кичик сони }
```

```
max: real;   { кетма-кетликнинг энг катта сони }
```

```
i: integer;  { цикл ҳисобчиси }
```

```
begin
```

```
writeln ('Қаср сонлар кетма-кетлигини қайта ишлаш.');
```

```
write ('Кетма-кетликдаги сонлар миқдорини киритинг —>');
```

```
readln (n);
```

```
writeln ('Кетма-кетликни киритинг.');
```

```
writeln ('Ҳар бир сон киритилганидан кейин <Enter>ни босинг');
```

```

write ('→ ');
readln (a); {кетма-кетликнинг биринчи сонини киритамиз}
      {фараз қиламиз:}
min:=a; {биринчи сон энг кичик}
max:=a; {биринчи сон энг катта}
sum:=a;
{қолган сонларни киритамиз}
for i:= 1 to n-1 do
  begin
    write ('→ ');
    readln (a) ;
    sum:=sum + a;
    if a < min then min:=a;
    if a > max then max:=a;
  end;
  o`rta: = sum/n;
  writeln ('Сонлар миқдори:', n);
  writeln ('Ўрта арифметик:', o`rta : 6:2);
  writeln ('Минимал сон', min:6:2);
  writeln ('Масимал сон:', max:6:2);
  writeln('Дастур ишини тугатиш учун <Enter>ни босинг ');
  readln;
end.

```

84 - масала

{Тасодифий сонларнинг учта кетма-кетлигини ҳосил қилади ва ҳар бир кетма-кетликнинг ўрта арифметигини ҳисоблайди}

const

```

n=3; {кетма-кетлик миқдори}
l=10; {кетма-кетлик узунлиги}

```

var

```

r: integer; {тасодифий сон}
sum: integer; {кетма-кетлик сонлари йиғиндиси}
o`rta: real; {ўрта арифметик}
i,j: integer; {цикл ҳисобчиси}

```

```

begin
  writeln ('Тасодифий сонлар');
  Randomize; { тасодифий сонлар генераторини инициализациялаш }
  for i:=1 to n do
    begin
      { кетма-кетлик ҳосил қилиш }
      sum:=0; { ноллаштириш! }
      for j:= 1 to l do
        begin
          r:=Random (10)+1;
          write (r:3);
          sum:=sum+r;
        o`rta:=sum/l;
        writeln ('Ўрта арифметик.: ', o`rta:6:2);
        end;
      writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
      readln;
    end.

```

85 - масала

```

  {  $y=|x|$  функция жадвали }
  const
    A = -4;      { аргумент ўзгариш соҳасининг куйи чегараси }
    B = 4;      { аргумент ўзгариш соҳасининг юқори чегараси }
    DX = 0-5;   { аргумент орттирмаси }
  var
    x,y: real;   { аргумент ва функция қиймати }
    n: integer;  { нуқталар миқдори }
    j : integer; { цикл ҳисобчиси }
  begin
    writeln ('  $y=|x|$  функция қийматлар жадвали');
    n:=Round ((B - A)/DX)+1;
    x:=A;
    for i:= 1 to n do
      begin

```

```

        y:=Abs (x);
        writeln (x:6:2,y:6:2) ;
        x: = x+DX;
    end;
readln;
end.

```

86 - масала

```

{7 га кўпайтириш жадвалини чиқариш}
var
m: integer; {кўпайтириш жадвали чиқариладиган сон (кўпаювчи) }
n: integer; {кўпайтувчи}
p: integer; {кўпайтма}
begin
m:=7;
for n:= 1 to 10 do
begin
p:=m*n;
writeln (m, 'x',n, ' = ',p);
end;
readln; {натижа экрандан йўқолмаслиги учун}
end.

```

87 - масала

```

{Ўнлик сонни иккилик сонга айлантириш}
var
o'n: integer; {ўнлик сон}
v: integer; {шаклланадиган рақам хонаси (разряди) салмоғи }
i: integer; {шаклланадиган рақам хонаси (разряд) тартиби қиймати
(номери)}
begin
writeln ('Ўнлик сонни иккиликка айлантириш');
writeln ('0 дан 255гача бўлган сонни киритинг ва <Enter>ни босинг');
writeln ('→');
readln (o'n);
write (O'n, 'ўнлик сонига иккилик мос келади'),

```

```

v:=128; {бош разряд салмоғи}
for i:= 1 to 8 do
begin
  if o'n >= v then
  begin
    write('1');
    o'n: = o'n - v;
  end
  else write ('0');
  v:= Round (v/2); {навбатдаги разряд салмоғи жорий разряд
                  салмоғидан икки марта кам}
end;
readln;
end.

```

88 - масала

```

{Экранга Пифагор квадратини–кўпайтириш жадвалини–чиқариш}
var
i,j: integer; {жадвал сатр ва устун рақами }
p: integer; { i нинг j га кўпайтмаси }
begin
write(' ': 4); {жадвал юқори чап бурчак катаги }
for j:= 1 to 10 do {устун рақами биринчи сатри }
write(j:4);
writeln;
for i: = 1 to 10 do
begin
write (i:4); {сатр рақами}
for j:= 1 to 10 do {жадвал сатри }
write(i*j:4);
writeln;
end;
readln; {натижа экрандан йўқолмаслиги учун}
end.

```

89 - масала

{1 -1/3 + 1/5 - 1/7 + ... қатор йиғиндисини ҳисоблаш }

var

x: real; { қатор ҳади }
n: integer; { қўшиладиган ҳадлар миқдори }
summ: real; { хусусий йиғинди }
i: integer; { цикл ҳисобчиси }

begin

writeln ('1 -1/3 + 1/5 - 1/7 + ... қатор йиғиндисини ҳисоблаш');
write ('Қаторнинг қўшиладиган ҳадлар миқдорини киритинг →');
readln (n);

for i:= 1 to n do

begin

 x:= 1/(2*i - 1);
 if (i mod 2) = 0
 then x:= -1*x;
 summ: = summ + x;

end;

writeln ('Қатор йиғиндисини:',summ:В:6) ;
writeln(pi/4:8:6, ' - pi/4 қиймати'); { p_i - номланган ўзгармас }
readln;

end.

90 - масала

{Кўпайтириш жадвалини билишни текшириш}

uses

Crt;

var

son1, son2:integer; { биринчи ва иккинчи сон }
nat:integer; { биринчи сонни иккинчи сонга кўпайтириш натижаси }
jav:integer; { Синалувчи жавоби }
miqdor:integer; { Тўғри жавоблар миқдори }
i:integer;

begin

 ClrScr; { экранни тозалаш }

```

writeln ('*** Кўпайтириш жадвалини билишни текшириш ***');
writeln ('Мисолдан кейин жавобни киритинг ва <Enter>ни босинг');
writeln;
miqdor:=0;          {тўғри жавоблар}
Randomize;         {тасодифий сонлар генераторини инициализациялаш}
for i:=1 to 10 do {10 та мисол}
begin
son1:=Random(9)+1;
son2:=Random (9) +1 ;
nat:=son1 * son2;
write (' ',son1,'x',son2,' =');
readln (jav);
if jav = nat
then
miqdor:= miqdor +1
else begin
writeln ('Адашдингиз!, ',son1, '*',son2, ' = ',nat) ;
end;
end;
writeln ('Тўғри жавоблар:', miqdor);
write ('Сизнинг баҳонгиз: ');
case miqdor of
10: writeln ('5');
9,8: writeln ('4');
7,6: writeln ('3');
0..5:writeln ('2');
end;
writeln (' Дастур ишини тугатиш учун <Enter>ни босинг');
readln;
end.

```

91 - масала

```

{Сонларни қўшиш ва айира билишни текшириш}
uses
  Crt;

```

var

```
son1, son2:integer; {сонлар}
amal:integer;      {Сонлар устида амаллар 0 - қўшиш, 1 – айириш}
belgi: char;       {амал белгилари - "плюс" ёки "минус"}
nat:integer;       {Натижа}
jav:integer;       {Синалувчи жавоби}
miqdor:integer;    {Тўғри жавоблар миқдори}
buf:integer;       { son1<son2 бўлган ҳолда son1 ва son2 ни алмаштириш
                    учун буфер}
i: integer; {цикл ҳисобчиси}
```

begin

```
ClrScr;
writeln ('Сонларни қўшиш ва айира билишни текшириш. ');
writeln ('Мисолдан кейин жавобни киритинг ва <Enter>ни босинг. '),
miqdor:=0;
Randomize;
for i:=1 to 10 do
  begin
    son1:=Random (9)+1           {биринчи сон}
    son2:=Random (9)+1           {иккинчи сон}
    amal:=Random (2)             {сонлар устида амаллар }
    if amal=0 then
      begin { қўшиш}
        nat: =son1 + son2;
        belgi:='+';
      end
    else
      begin { айириш}
        belgi:= ' - ';
        if son1 < son then
          begin { son1 ва son2 ни алмаштирамиз}
            buf: = son2 ;
            son2 :=son1 ;
            son1 :=buf;
          end;
      end;
  end;
```

```

nat :=son1-son2 ;
end;
write ( ' , son1, belgi, son2, '=' );      { мисолни чиқариш }
readln (jav) ;                             { синовчи жавобини ҳосил қилинг }
if jav = nat
then
miqdor:= miqdor +1
else
begin
writeln ('Адашдингиз! ' , son1 , belgi, son2 , '=' , nat );
end;
end;
writeln ('Тўғри жавоблар: ' , miqdor) ;
write ('Баҳонгиз:');
case miqdor of
10: writeln ('5');
9,6: writeln ('4');
7,6: writeln ('3');
0..5: writeln ('2');
end;
writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
readln;
end.

```

92 - масала

```

{Оддий электрон соатлар}
uses Crt; {Crt кутубхонаси процедура ва функцияларидан фойдаланиш}
var
min,sec: integer; {минутлар, секундлар}
begin
ClrScr; { экранни тозалаш }
for min:= 1 to 3 do
begin
for sec:=1 to 60 do
begin

```

```

Delay (1000);      { 1000 ms тўхтатиб туриш }
GotoXY (1,1); { курсорни 1-сатрнинг 1-устунига ўрнатиш }
write (min,':',sec, ' ');
if KeyPressed {агар клавиш босилган бўлса}
    then Halt;      {дастурни тугатиш}
end;
end;
end.

```

3.1.6. REPEAT КўРСАТМАЛИ ДАСТУРЛАР

97 - масала

```

{Клавиатурадан киритилган бутун соннинг оддийлигини текшириш}
var
n: integer;
d: integer;
r: integer;
begin
write ('Бутун сонни киритинг → ');
readln(n);
d:= 2; {аввал иккига бўламиз }
repeat
r:=n mod d;
if r<>0 {n dга қолдиқсиз бўлинмади}
then d:=d + 1;
until r=0; {ҳозирча n бўлинадиган сонни топмадик}
if d=n
then writeln (n,' – туб сон. ')
else writeln (n,' – туб сон эмас. ');
end.

```

98 - масала

```

{ «Сонни топ» ўйини }
const
urinish = 5; {ўйинчига бериладиган уринишлар сони}
var

```

```
comp: integer; { компьютер «ўйлаган» сон }
o'yinchi: integer; { ўйинчи варианты }
n: integer; { ўйинчининг уринишлар сони }
```

begin

```
Randomize; { тасодикий сонлар генераторини инициализациялаш }
comp:=Random(9)+1; { компьютер сон ўйлади }
writeln (' «Сонни топ» ўйини');
writeln ('Компьютер 1дан 10 гача сон «ўйлади»');
writeln ('уни', urinish,' уринишда топинг. ');
writeln ('Сонни киритинг ва <Enter>ни босинг.');
```

repeat

```
n: = n + 1;
write ('→ ');
readln (o'yinchi);
until (n = urinish) or (comp = o'yinchi);
if comp = o'yinchi
    then writeln ('Ютдингиз!')
    else writeln ('Ютказдингиз! Компьютер', comp, 'сонини ўйлаган эди');
readln;
```

end.

99 - масала

```
{ Таймер }
```

```
uses Crt;
```

```
var
```

```
min,sec: integer; { тўхтатиш: минут ва секунд }
```

begin

```
writeln ('Минут ва секундларда тўхтатиш миқдорини киритинг')
writeln ('Масалан 2 30 ва <Enter>ни босинг. ');
write ('→ ');
readln (min,sec);
ClrScr;
GotoXY(2,2);
write (min,':',sec,' ');
repeat
```

```

if sec = 0 then
begin
    min:=min-1;
    sec:=60;
end;
repeat
    Delay (1000); {тўхтатиш 1 сек}
    sec:= sec-1;
    GotoXY(2,2);
    write (min,',',sec,' ');
    if KeyPressed {агар клавиш босилган бўлса }
        then Halt;          {дастурни якунланг }
until sec =0;
until (min = 0) and (sec = 0) ;
    {товушли сигнали }
    Sound (1000);          {частотаси 1000 герцга тенг бўлган товуш
                           сигнални кўшинг}
    Delay (500);
    Nosound;              {товушни ўчиринг }
end.

```

3.1.7. WHILE КўРСАТМАЛИ ДАСТУРЛАР

103 - масала

```

    { ПИ сонини ҳисоблаш)
var
    p:real;          {"ПИ"нинг ҳисобланадиган қиймати }
    a:real;          {ҳисоблаш аниқлиги}
    n:integer;       {қатор ҳади рақами}
    elem:real;       {қатор ҳади қиймати}
begin
    p:=0;
    n:=1;
    elem:=1;        {бошланғич қиймат}
    write (' ПИ ҳисоблаш аниқлигини беринг → ');
    readln (a);

```

```
writeln (' ПИ ни қуйидаги аниқлик билан ҳисобланг', a:9:6);
while elem >= a do
begin
    elem:=1/(2*n-1);
    if (n MOD 2) = 0
    then p:=p-elem
    else p:=p+elem;
    n:=n+1;
end;
p:=p*4;
writeln (a:9:6, 'аниқликдаги ПИ қиймати ', p:9:6, ' га тенг')
writeln ('Қаторнинг, 'n', ҳади қўшилган. ');
readln;
end.
```

104 - масала

{Икки бутун соннинг энг катта умумий бўлувчисини топиш }

var

```
n1,n2:integer;      {ЭКУБ топилаётган сонлар}
ekub:integer;       {энг катта умумий бўлувчи}
r:integer;          {n1 ни n2 га бўлишдаги қолдиқ}
```

begin

```
writeln ('Икки бутун сон учун энг катта умумий бўлувчини ҳисоблаш');
write ('Бир қатордан иккита сонни киритинг ');
writeln ('ва <Enter>ни босинг');
write('→');
readln (n1,n2);
while (n1 mod n2) <>0 do
begin
    r:=n1 mod n2;      {бўлишдаги қолдиқ}
    n1:=n2;
    n2:= r;
end;
ekub:=n2;
writeln (n1,' ва ',n2,' сонларининг ЭКУБ : ', ekub);
```

```
readln;  
end.
```

3.1.8. БЕЛГИЛАР ВА ҚАТОРЛАР ИШЛАТИЛГАН ДАСТУРЛАР

109 - масала

```
{Саломлашиш }  
var  
ism: string [40];      {фойдаланувчи исми }  
begin  
  writeln ('Исмингиз?');  
  writeln ('(Исмингизни киритинг ва <Enter>ни босинг');  
  write ('→ ');  
  readln (ism);  
  writeln ('Салом, ', ism, ' ! ');  
  readln;  
end.
```

110 - масала

```
{Ахборотни белгилар бўйича чиқариш }  
uses  
crt;      { Delay процедурасини ишлатиш учун }  
var  
msg: string [80];      {ахборот }  
n; integer;            {чиқариладиган белги рақами }  
begin  
  msg:= 'Буюк дастурчига салом! ';  
  for n:=1 to Length (msg) do  
    begin  
      write (msg[n]);  
      Delay (100);      { 0.1 секундга тўхтатиш }  
    end;  
  readln;  
end.
```

111 - масала

```
{Киритилган белги кодини ASCII стандартида чиқариш}
var
belgi; char;    {белги}
code: integer; {белги коди }
begin
writeln (' белгини киритинг ва <Enter>ни босинг. ');
writeln ('Дастур ишини тугатиш учун нуқтани киритинг. '),
repeat
write ('→ ');
readln (belgi);
code:= Ord (belgi);
writeln ('Белги: ',belgi,' Код: ',code);
until belgi='.';
end.
```

112 - масала

```
{Белгиларни кодлаштириш жадвалини чиқариш }
var
ch:char;        {белги}
o'n: integer,   {белгининг ўнлик коди }
i,j: integer;
begin
o'n:=0;
for i:=0 to 15 do {ўн олти сатр }
begin
o'n:=i; {128-255 кодлар белгилари учун кодлаштириш
жадвалини ҳосил қилишда бу кўрсатмани o'n:=i+128 га
алмаштириш керак;}
for j:=1 to 8 do {саккизта устун }
begin
if (o'n <7) or (o'n >=14)
then
write(o'n:4,'-',',',
chr (o'n):1, chr (179))
```

```

else {CR,LF,TAB белгилари ифодаланмайди }
    write (o'n:4,'-',chr (179));
    o'n:= o'n +16;
end;
writeln { экраннинг янги сатрига ўтиш };
end;
readln;
end.

```

113 - масала

```

var
    satr: string [80];           { матн сатри }
    uzun: integer;              { сатр узунлиги }
    i: integer;                  { қайта ишланадиган белги рақами }
begin
    writeln ('Матн сатрини киритинг ва <Enter>ни босинг');
    writeln ('→');
    readln (satr);
    uzun:= Length (satr);
    for i:=1 to uzun do
        case satr [i] of
            'a' .. 'n': satr [i]:= chr (ord (satr [i]) - 32);
            'p' .. 'я': satr [i]:= chr (ord (satr [i]) - 80);
        end;
    writeln ('Юқори регистрга айлантирилган сатр:'),
    writeln (satr);
    readln ;
end.

```

114 - масала

```

{ Сатрнинг бошланғич бўшлиқларини олиб ташлаш }
var
    satr: string [80];           { сатр }
begin
    writeln ('Сатрнинг бошланғич бўшлиқларини олиб ташлаш. ');

```

```

write ('Сатрни киритинг:');
readln (satr);
while (pos(' ', satr) = 1) and (length (satr) > 0) do
    delete (satr,1,1);
write ('Бошланғич бۆшликларсиз сатр:' , satr),
readln;
end.

```

115 - масала

{Клавиатурадан киритилган соннинг бутунлигини текшириш}

var

```

satr: string [40];           {сатр}
n: integer;                 {текшириладиган белги рақами}

```

begin

```

writeln ('Сонни киритинг ва <Enter>ни босинг');
write ('→ ');
readln (satr);
n:=1;
while (n <= Length(satr)) and
    ((satr [n] >= '0') and (satr [n] <='9'))

```

```

    do n:=n+1;

```

```

write ('Киритилган сатр ');

```

```

if n < Length (satr)

```

```

    then write ('бутун сон бўлмади ');

```

```

writeln ('бутун сон бўлади.');
```

```

readln;
```

end.

116 - масала

{Киритилган сатрнинг ўн олтилик сон эканлигини текшириш}

var

```

satr: string [20];         {сатр }
i: integer;                {текширилаётган белги рақами}
error: boolean; {сатрда хато белги бор}

```

begin

```

writeln ('Ўн олтилик сонни киритинг ва <Enter>ни босинг') ;
write ('→');
readln (satr);
{ киритилган сатрни юқори регистрга алмаштирамиз }
for i:=1 to Length (satr)
do satr [i] :=UpCase (satr [i]) ;
i:=1;
error:=FALSE;
while (i <= Length (satr)) and (not error)
do if ((satr [i] >= '0') and (satr [i] <= '9')) or
((satr [i] >= 'A') and [satr [i] <= 'F'))
then i:= i+1
else error:=TRUE;
write ('сатр ');
if error
then write ('Ўн олтилик сон бўлмади');
writeln ('Ўн олтилик сон бўлади. ');
readln;
end.

```

117 - масала

```

{ Киритилган сатр ишорасиз каср сон эканлигини текширади }
var
satr: string [20];      { сатр }
i: integer;             { текширилаётган белги рақами }
err: boolean;          { TRUE – сатр каср сон эмас }
begin
writeln ('Каср сонни киритинг ва <Enter>ни босинг');
write ('→ ');
readln (satr);
i:=1;
err:=TRUE; { сатр каср сон бўлмасин }
if (satr [i] >='1') and (satr [i] <= '9') then { рақамнинг биринчи белгиси }
begin
{ рақамдан кейин яна рақам келиши мумкин }

```

```

while (satr [i] >='1') and (satr [i] <= '9') and (i<Length(satr))
  do i:= i +1;
  {рақамлардан кейин нуқта келади, лекин у охирги белги эмас}
  if ((satr [i] = ' . ') and (i < Length(satr))) then {нуқта}
    begin
      i:= i + 1;
      {нуқтадан кейин ҳеч бўлмаганда битта рақам бўлиши
      керак}
      if ((satr [i] >='1') and [satr [i] <='9']) then
        begin
          while ((satr [i] >='1') and (satr [i] <:='9') and
            (i<Length(satr)))
            do i:= i + 1;
          if i= Length(satr) {рақамнинг охирги белгиси}
            then err:=FALSE; {хато тўғрисидаги фараз ёлгон}
        end;
      end;
    end;
  write ('Сатр ');
  if err
    then write('каср сон бўлмади ');
  writeln ('каср сон бўлади. ');
  readln;
end.

```

118 - масала

```

{Клавиатурадан киритилган саккиз хонали иккилик сонни ўнликка
айлантириш}
var
  ikki: string [8];           {иккилик сонини ифодалаш }
  o'n: integer;              {ўнлик сон}
  r: string [1];            {иккилик сонининг i-чи хонаси}
  s: integer;                {иккилик сонинг i-хона салмоғи}
  i: integer;                {иккилик сон хонасининг рақами}
begin

```

```

writeln ('Саққиз хоналик иккилик сонини киритинг');
writeln ('ва <Enter>ни босинг. ');
write ('→ ');
readln (ikki);
if Length (ikki) <> 8 then
    writeln ('Сон саққиз хонали бўлиши керак.')
else begin
    o'n := 0 ;
    s:=128; {бош (8) хоналик иккилик сонининг салмоғи}
    for i:=1 to 8 do
        begin
            r:=ikki[i]; {i-хонани ажратинг}
            if r = '1' then
                o'n:=o'n +s;
                s:=Round (s/2) ; {навбатдаги хона салмоғини ҳисобланг}
        end;
        write (ikki, 'Иккилик сонига ', o'n, 'ўнлик сон мос келади');
end;
writeln ('Дастур ишини тугатиш учун <Enter>ни босинг. ');
readln;
end.

```

119 - масала

```

var
    satr: string [2];      {ўн олтилик сон }
    d: integer;           {киритилган ўн олтилик сонга мос келувчи ўнлик сон}
    s: integer;           {ўн олтилик сон хонасининг салмоғи}
    i: integer;
begin
    writeln ('икки хонали ўн олтилик сонни киритинг', 'ва <Enter>ни
        босинг. ');
    write ('→');
    readln (satr);
    {киритилган сатрни юқори регистрга айлантирамиз }
    for i:=1 to Length (satr)

```

```

do satr [i]: = UpCase (satr [i]);
i = Length (satr);           { кичик хонадан қайта ишлаймиз }
s: = 1;                       { ва унинг вазни 1 га тенг бўлади }
while (i > 0) and
  (((satr [i] >='0') and (satr [i] <= '9')) or
   ((satr [i] >= 'A') and (satr [i] <= 'F'))) do
begin
  { бу ерда – белги рақам ёки А дан F гача лотин ҳарфлари }
  if (satr [i] >= '0') and (satr [i] <= '9')
  then d:=d+s*(Ord (satr [i] – 48)
       { Ord ('0') = 48, Ord ('1') = 49, и т.д. }
  else d:=d+s*(Ord (satr [i] – 55);
       { Ord ('A') = 65, Ord ('B') = 66, и т.д. }
  i:= i-1;           { олдинги хонага }
  s:=s*16;
end;
if i=0   { ҳамма хоналар ишлаб чиқилган }
then writeln (satr, 'Ўн олтилик сонга ўнлик ', d, ' сони мос келади')
else writeln ('Кирилган сатр ўн олтилик рақам бўлмайди. ');
readln;
end.

```

120 - масала

```

{ Ўнлик сонни фойдаланувчи кўрсатган санок системасига (иккиликдан
  ўнликкача) ўтказиш }
uses Crt;
var
  a:integer;   { Санок система асоси }
  n:integer;   { Бошланғич сон
  r:integer;   { Сонни санок системаси асосига бўлгандаги чиққан
               қолдиқ. }
  s:string [16]; { Сонни берилган санок системасида ифодалаш. }
  buf:string [1];
begin
  ClrScr;

```

```
Write ('Бутун сонни киритинг → ');  
readln (n);  
write ('Санок системаси асосини киритинг → ');  
readln (a);
```

{Бошланғич сонни санок система асосига бўлишдан чиққан қолдиқни система асосидан катта бўлгунга қадар бўламиз. Ҳар бир кадамдаги бўлиш қолдиғи – навбатдаги сон }

repeat

```
r:=n mod a; { навбатдаги рақам }  
n:=n div a; { бўлишнинг бутун қисми }  
Str (r,buf); { рақамни сатрга айлантириш }  
s:=buf+s;
```

until (n<0);

Str (n,buf);

writeln(s);

readln;

end.

121 - масала

{ Ўнлик сонни ўн олтиликка айлантириш }

uses Crt;

var

n:integer; { Бошланғич сон }

r:integer; { Сонни санок системасининг асосига бўлгандаги қолдиқ }

s:**string** [16]; { Сонни берилган санок системада ифодалаш }

buf: **string** [1];

begin

Clr Scr

write (' Бутун сонни киритинг → ');

readln (n);

s:= ' ';

{ Берилган сонни санок сон системаси асосига (16), бўлишдан чиққан қолдиқ система асосидан катта бўлгунга қадар, бўламиз. Ҳар бир кадамдаги бўлиш қолдиғи – навбатдаги сон }

```

write (n, 'Ўнлик сонга ўн олтилик сон мос келади');
repeat
  r:=n mod 16;           {навбатдаги рақам};
  n:= n div 16;         {бўлишнинг бутун қисми};
  if r<0
  then buf:= chr (r+48)  {chr(48)= '0', chr (49) = '1' ва ҳоказо }
  else buf:= chr (r+55)  {chr(65)= 'A', chr (66) = 'B' ва ҳоказо }
  s:=buf + s;
until (n<16)
if n <> 0 then begin
  if n<0
  then buf:= chr (n+48)
  else buf:= chr (n+55);
  s:=buf + s;
end;
writeln (s);
readln;
end.

```

122 - масала

{Дастур клавиатурадан киритилган арифметик ифода қийматини ҳисоблайди}

```
uses Crt;
```

```
var
```

```

s: string [80];       {сатр}
l: integer;           {сатр узунлиги }
к: integer;           {ифода қиймати }
n: integer;           {навбатдаги сон }
i: integer;           {сатрдаги сон рақами}
err: integer;         {белгини сонга айлантиришдаги хатолик коди }

```

```
begin
```

```

ClrScr;
writeln ('Арифметик ифодани киритинг,');
writeln ('масалан, 4+5-3-5+2 ва <Enter>ни босинг');
write ('→ ');

```

```

readln (s);
val (s[1],k,err);
i:=3;
repeat
    val (s[i],n,lrr); { навбатдаги бир хонали сонни ҳосил қилинг }
    if s [i-1] = ' +'
        then k:=k+n
        else k:= k-n;
    i:=i+2;
until i > Length(s);
writeln ('Киритилган ифода қиймати: ',k);
writeln ('Дастур ишини якунлаш учун <Enter>ни босинг. ');
readln;
end.

```

3.1.9. МАССИВЛАРГА ОИД ДАСТУРЛАР

123 - масала

```

{Массивнинг ноламас элементлар миқдорини ҳисоблаш}
const
    SIZE=5; { массив ўлчами }
var
    a: array [1 .. SIZE] of integer;
    n: integer; { ноламас элементлар миқдори }
    i: integer; { индекс }
begin
    writeln ('Бутун сонлар массивини киритиш. ');
    write ('Ҳар бир сон киритилгандан сўнг ');
    writeln (' <Enter>ни босинг ');
    n:=0;
    for i:= 1 to SIZE do
        begin
            write ('a [',i,'] → ');
            readln (a[i]);
            if a[i] <> 0 then n:= n+1;
        end;

```

```
writeln ('Массивда ', n та,'нолмас элемент бор.');
```

```
readln;
```

```
end.
```

124 - масала

```
{Массивнинг минимал элементини топиш }
```

```
const
```

```
n=5; {массив ўлчами}
```

```
var
```

```
a: array [1 .. n] of integer;
```

```
min: integer; {массив минимал элементи}
```

```
i : integer;
```

```
begin
```

```
writeln ('Массивнинг минимал элементини топиш.');
```

```
writeln ('Бир сатрдан массивнинг бутун сонлар элементларини')
```

```
writeln (n', киритинг ва <Enter>ни босинг.');
```

```
write ('→ ');
```

```
for i:=1 to n -1 do
```

```
    read (a[i]);
```

```
readln (a[n]);           {дастурнинг охиридаги readln кўрсатма тўғри  
                          ишлаши учун }
```

```
min:= a[i];             {фараз қиламиз, биринчи элемент минимал }
```

```
{массивнинг қолган элементларини минимал билан таққослаймиз }
```

```
for i:=2 to n do
```

```
    if a[i] < min then min:= a[i];
```

```
writeln ('Массивнинг минимал элементи: ',min);
```

```
readln;
```

```
end.
```

125 - масала

```
{Ҳавонинг ҳафталик ўртача ҳароратини ҳисоблаш }
```

```
const
```

```
{ҳафта кунлари – сатрли ўзгармаслар массивидир }
```

```
day: array [1..7] of string [11] = ('Душанба','Сешанба',  
                                     'Чоршанба','Пайшанба','Жума',  
                                     'Шанба','Якшанба');
```

```
var
```

```
temper:array [1..7] of real; {харорат}  
sum:real;           {хафталик харорат йиғиндиси}  
o`rta: real;       {хафталик ўртача харорат}  
i:integer;
```

```
begin
```

```
writeln ('Хафталик ҳаво хароратларини беринг. ');
```

```
for i:=1 to 7 do
```

```
begin
```

```
write (day [i],'→ ');
```

```
readln (temper[i]);
```

```
end;
```

```
{Хафталик ўртача хароратни ҳисоблаш}
```

```
sum:=0;
```

```
for i:=1 to 7 do
```

```
sum:=sum + temper [i];
```

```
o`rta:=sum/7;
```

```
writeln ('Ўртача хафталик харорат:',o`rta:6:2);
```

```
readln;
```

```
end.
```

126 - масала

{Массивда элементларни кўриб чиқиш усули билан излаш}

```
var
```

```
massiv:array [1..10] of integer; {бутун массив }
```

```
namuna:integer;           {текстда излаш учун намуна }
```

```
topildi:boolean;        {намуна билан мослик белгиси }
```

```
i:integer;
```

```
begin
```

```
{10 та бутун сонни киритиш }
```

```
writeln ('Массивда излаш.');
```

```
write ('Бир сатрдан 10 та сонни бўшлиқ орқали киритинг');
```

```

writeln (' ва <Enter>ни босинг');
write ('→ ');
for i:=1 to 10 do read (massiv[i]);
{сонлар массивга киритилди }
write ('Излаш учун намуна киритинг (бутун сон)→ ');
readln (namuna);
{оддий кўриб чиқиш билан излаш }
topildi:=FALSE;    {мослик йўқ }
i:=1; { массивнинг биринчи элементидан бошлаб текширамыз}
repeat
    if massiv[i] = namuna
        then topildi:=TRUE {намуна билан мос тушди}
        else i:=i+1; {навбатдаги элементга ўтиш}
    until (topildi) or (i>10);    {агар намуна билан мослик бўлса, ёки
                                     массивнинг охириги элементи текширилса,
                                     иш тугайди }
    if topildi
        then writeln ('Намуна билан мос тушган элемент рақами', i:3, '. ')
        else writeln ('Намуна билан мослик йўқ. ');
    readln;
end.

```

132 - масала

{Массив элементлари ўсиб боровчи кетма-кетлик ҳосил қилишини текшириш)

const

n =5;

var

a:array [1. . n] **of** integer; {массив }

k:integer; {индекс }

ok:boolean; {ok = True – камаймайдиган кетма-кетлик }

begin

writeln ('Элемент массивлари ўсиб боровчи кетма-кетлик');

writeln ('ҳосил қилишини текшириш');

write ('Бир сатрдан бутун сонларни', n, 'киритинг');

```

writeln ('ва <Enter>ни босинг');
for k:=1 to n -1 do
  read(a[k]);
readln (a[n]);
k:=1;
ok:=TRUE;
repeat
  if a[k] < a[k+1]
    then k:=k+1
    else ok:=FALSE;
until (ok =FALSE) or (k = n);
write ('Киритилган кетма-кетлик');
if not ok
  then write('ўсиб бормайдиган кетма-кетлик');
writeln ('ўсиб борувчи кетма-кетлик. ');
readln;
end.

```

133 - масала

```

{Клавиатурадан киритилган сон массивда неча марта учрашишини
текшириш}
const
  m=5;
var
  a: array [1.. m] of integer; {массив}
  nam: integer;               {намуна}
  n: integer;                 {қиймати намунага тенг массивдаги элементлар
                               миқдори}
  i: integer; {индекс}
begin
  write (' Массивни киритинг - ');
  writeln (m,' бир сатрдаги бутун сонлар. ');
  write ('→');
  for i:=1 to m-1 do
    read(a[i]);

```

```

readln(a[m]);
write ('Таққослаш учун намуна киритинг → ');
readln (nam);
n:=0;
{қайта ишлаш}
for i:=1 to m do
  if a[i] = nam then n:=n+1;
  if n<> 0
    then writeln (nam, 'сони массивда ', n, 'марта учрайди')
    else writeln ('Массивнинг биронта элементи намунага тенг
    эмас. ');
  readln;
end.

```

139 - масала

```

{Массивни тўғридан-тўғри танлаш усули билан ўсиб боришини
саралаш}
const
  n =5;           {Массив ўлчами}
var
  a:array [1 .. n] of integer;
  i:integer;      {минимал элементни излаш бошланган элемент рақами}
  min:integer;    {массивнинг i дан юқори чегарасигача соҳадаги элемент
рақами}
  j: integer;     {минимал элемент билан таққосланадиган элемент
рақами }
  buf:integer;    {массив элементларини алмаштиришда ишлатиладиган
буфер}
  k: integer;     {киритиш, чиқариш учун индекс }
begin
  writeln ('Массивни саралаш. ');
  write ('Бир сатрдан ', n, 'та бутун сонларни киритинг');
  writeln ('ва <Enter>ни босинг');
  write ('→ ');
  for k:=1 to n-1

```

```

do read(a[k]);
readln (a[n]);
writeln ('Саралаш');
for i:=1 to n-1 do
begin
    {массивнинг a[i] дан a[n] гача қисмида минимал элементни излаш}
    min:=i;
    for j:=i + 1 to n do
        if a[j]<a[min] then min: = j;
        { a [min] ва a[i] ларнинг ўрнини алмаштириш }
        buf:= a[i];
        a[i]:=a[min];
        a[min]:=buf;
        { саралаш цикли тугади }
        { массивнинг оралик ҳолатини чиқариш }
    for k:=1 to n do write (a[k], ' ');
    writeln;
end;
writeln;
{ сараланган массивни чиқариш }
writeln (' Массив сараланган. ');
for k:=1 to n do
    write (a[k], ' ');
readln;
end.

```

140 - масала

```

{«Кўпиклар» усули билан массив элементларини ўсиб бориш тарзида
тартибга солиш }
const
n=5;
var
a:array [1.. n] of integer;
i:integer; {Цикл ҳисобчиси}
k:integer; {Массив элементининг жорий индекси}

```

```

    buf:integer;
begin
    writeln (' «Кўпик» усули билан массивни саралаш. ');
    write ('Бир сатрдан ', n, ' та бутун сонни киритинг' )
    writeln ('ва <Enter>ни босинг');
    for k:=1 to n-1 do
        read(a[k]);
    readln (a[n]);
    writeln ('Саралаш. ');
    for i:=1 to n-1 do
        begin
            for k:=1 to n-1 do
                begin
                    if a[k] > a[k+1] then
                        begin
                            { к элементни (k+1) элементга алмаштириш }
                            buf:=a[k];
                            a[k]:=a[k+1];
                            a[k+1]:=buf;
                        end;
                    end;
                { Навбатдаги саралашдан кейин массив ҳолатини чоп этиш }
                for k:=1 to n do write (a[k], ' ');
                writeln;
            end;
        writeln;
        writeln ('Массив сараланди. ');
        for k:=1 to n do write(a[k], ' ');
        readln;
    end.

```

141 - масала

```

    {Иккита тартибга солинган массивларни бирлаштириш }
const
    n = 5;           { бошланғич массивлар ўлчами }

```

var

a,b: **array** [1.. n] **of** integer; { бошланғич массивлар }

c; **array** [1..2*n] **of** integer; { натижа - массив }

k,l,m: integer; { a,b ва c массив индекслари }

begin

writeln ('Иккита тартибга солинган массивларни ўсиб боришига кўра
бирлаштириш.');

writeln ('Битта сатрдан биринчи элемент массивларини киритинг');

write ('(', n, ' та бутун сонни') → ');

for k:= 1 **to** n-1 **do** read (a[k]);

readln (a [n]);

writeln ('Битта сатрдан иккинчи массив элементларини киритиш');

write ('(', n, ' та бутун сонни') →');

for l:=1 **to** n-1 **do** read (b[l]);

readln (b[n]);

k:=1; l:=1; m:= 1;

repeat

if a[k] < b[l]

then begin c[m] :=a[k];

m:=m+1;

k:=k+1;

end

else if a[k] > b[l]

then begin c[m]:=b[l];

m:=m+1;

l:=l+1;

end

else begin c[m] :=a[k]

c[m+1]:=b[l];

k:=k+1;

l:=l+1;

m:=m+2;

end;

until (k>n) **or** (l>n); {иккита бошланғич массивлардан биттаси С
массивга тўла ёзилди }

```

while k <= n do { A нинг C га ёзилмаган элементлари бор}
begin
  c[m] :=a[k] ;
  k:=k+1;
  m: =m+1;
end;
while l <= n do {B нинг C га ёзилмаган элементлари бор}
begin
  c[m]:=b[l];
  l:=l+1;
  m:=m+l;
end;
writeln ('Натижа - массив');
for i:=1 to 2*n do
  write [c[i],' ');
writeln;
writeln ('Дастур ишини тугатиш учун <Enter>ни босинг.')
readln;
end.

```

142 - масала

```

{Тартибга солинган массивда бинар излаш}
label 5;
const m=10;
var
  a:array[1..10] of integer; {бутун сонлар массиви}
  nam: integer;             {излаш учун намуна}
  ok: boolean;              {TRUE - массив тартибга солинган}
  o`rta,yuqori,quyi:integer; {массивнинг ўрта, юқори ва қуйи
                             элементлари рақами}
  found:boolean;           {намуна билан мос келиш белгиси}
  n:integer;               {намуна билан мослик ҳисобчиси}
  i:integer;
begin
  { массивни киритиш }

```

```

writeln ('*** Тартибга солинган массивда бинар излаш ***');
write ('Массивни киритинг (бир сатрдан', m, ' та бутун сонни');
writeln (' ва <Enter>ни босинг');
write ('→ ');
for i:=1 to m-1 do read(a[i]);
readln (a[m]);
{Массив тартибга солинганини текшириш}
i:=1;
ok:=TRUE; { массив тартибга солинган бўлсин}
while (i < HB) and ok
    do if a[i] <= a [i+1]
        then i:=i+1
        else ok:=FALSE;
if not ok then
begin
    writeln ('Киритилган массив ўсиб бориши бўйича тартибга
            солинмаган');
    goto 5;
end;
write ('Излаш учун намуна (бутун сон) киритинг → '),
readln (nam);
{бинар излаш}
yuqori:=1;
quyi:=HB;
found:=FALSE;
n:=0;
repeat
    o'rta:=(quyi-yuqori) div 2 + yuqori; { массивни тенг иккига бўламиз}
    n:= n+1;
    if a [o'rta] = nam
        then found:=TRUE
        else
            {изланаётган элемент қайси қисмда, юқори ёки пастда,
            бўлишини аниқлаш}
            if nam < a[o'rta]

```

```

        then quyi:= o`rta -1
        else yuqori:= o`rta +1;
until (yuqori > quyi) or found;
if found
then write (o`rta, '-рақамли элемент билан мос тушиш', n, ' та
        таққослаш олиб борилди.')
else writeln ('Массивда намуна топилмади.');
```

5: readln;

end.

143 - масала

```

{Талабалар бўйи ҳақидаги маълумот таҳлили}
const
max=30; {синфдаги ўқувчиларнинг энг кўп сони}
var
bo`y: real; {ўқувчи бўйи }
guruh: array [1.. max] of real; {ўқувчилар бўйи }
n: integer; {маълумотлари киритилган ўқувчилар сони }
o`rta: real; {ўрта бўй}
m: integer; {бўйи ўртадан юқори ўқувчилар сони }
sum: real; {жами бўй}
i: integer;
begin
writeln ('*** Ўқувчилар бўйининг таҳлили ***•');
writeln ('Бўй узунлиги (см) ни киритинг ва <Enter>ни босинг');
writeln (' Ишни тугатиш учун 0 ни киритинг ва <Enter> ни босинг');
n:=0;
repeat
write ('→ ');
readln (bo`y);
if bo`y <> 0 then
begin
n:=n+1;
guruh [n]:= bo`y;
sum:=sum+ bo`y;
```

```

    end;
until (n > max) or (bo'y = 0);
if n > 0 then
    begin
        o'rta: = sum/n;
        m:=0;
        {хар бир бўйни ўртача билан таққослаймиз}
        for i:=1 to n do
            if guruh [i] > o'rta then m:=m+1;
            writeln ('Ўртача бўй: ', o'rta;6:1,' см');
            writeln (m,'-та ўқувчининг бўйи ўртачадан баланд');
        end
    else writeln ('Қайта ишлаш учун маълумотлар
        йўқ. ');
    readln;
end.

```

144 - масала

{Устунлар бўйича массив элементларининг йиғиндиларини ҳисоблаш}

```

const
    m =3; {сатрлар сони }
    n =5; {устунлар сони}
var
    a: array [1.. m, 1.. n] of integer; {массив}
    s: array [1.. n] of integer; {элементлар йиғиндиси}
    i,j: integer;
begin
    writeln ('Массивни киритинг. ');
    writeln ('Ҳар бир, ' n, 'сатрдаги бутун сон элементларини', 'киритгандан
        кейин <Enter>ни босинг');
    for i:=1 to m do {m сатр}
        begin
            write('→');
            for j:=1 to n -1 do
                read (a[i,j]);

```

```

        readln (a[i, n]);
    end;
    writeln (' Киритилган массив');
    for i:=1 to m do
        begin
            for j:=1 to n -1 do
                write (a[i,j]:4);
                writeln (a[i, n]:4);
            end;
        {Қайта ишлаш}
        for j:=1 to n do {ҳар бир устун учун битта устун элементларини}
            {қўшамиз}
            for i:=1 to m do
                s[j]:=s[j] + a [i,j];
                writeln ('-----');
            for i:=1 to n do
                write (s[i]:4);
            writeln;
            readln;
        end.

```

145 - масала

```

{Иккинчи тартибли матрица аниқловчисини ҳисоблаш }
var
    a: array [1..2,1..2] of real;
    det: real;      { аниқловчи (детерминант) }
    i,j: integer;  { массив индекслари }
begin
    writeln ('Иккинчи тартибли матрицани киритинг. ');
    writeln ('Сатр элементлари киритилгач, <Enter> ни босинг');
    for i:=1 to 2 do
        begin
            write ('→');
            read (a[i,1]);
            readln (a[i,2]);

```

```

end;
det:=a[1,1]*a[2,2] - a[1,2] * a[2,1];
writeln (' Матрица аниқловчиси ');
for i:=1 to 2 do
  begin
    for j:=1 to 2 do
      write (a[i,j]:6:2);
      writeln;
    end;
  writeln (det:6:2,'га тенг');
  readln;
end.

```

146 - масала

```

{Матрица афсункор квадрат бўлишини текширинг};
const
  MAX = 5;      {матрицанинг максимал ўлчами}
var
  a: array [1..MAX, 1..MAX] of integer; {матрица }
  n: integer;   {текширилаётган матрица ўлчами}
  ok:boolean;   {TRUE - матрица афсункор квадрат бўлади}
  i,j: integer; {массив индекслари}
  sum: integer; {матрица бош диагонали элементлари йиғиндиси}
  s: integer;   {жорий сатр, устун ёки матрица иккинчи диагонали
                элементларининг йиғиндиси}
begin
  write ('Матрица элементларини киритинг (3..4) ',
        'ва <Enter>ни босинг → ');
  readln(n);
  writeln (' Матрица сатрларини киритинг');
  writeln (n,'сатрнинг бутун сонларини киритгандан кейин <Enter>ни
          босинг'),
  for i:=1 to n do
    begin
      write ('→')

```

```

for j:=1 to n-1 do read (a[i, j]) ;
readln (a[i,n]);

end;
ok:=TRUE;    { матрица афсункор квадрат бўлсин }
sum:=0;
{ бош диагонал элементлари йиғиндисини ҳисоблаймиз }
for i:=1 to n do
    sum:=sum +a[i, i] ;
    { сатрлар бўйича йиғиндиларни ҳисоблаймиз }
    i:=1;
repeat
    s:=0; { жорий сатр элементлари йиғиндиси }
    for j:=1 to n do s:=s+a [i, j];
    i:=i+1;
    if s <> sum then ok:=FALSE;
until (not ok) or (i > n);
if ok then
    {бу ерда ҳар бир сатр элементлари йиғиндиси бош диагонал
    элементлари йиғиндисига тенг }
begin
    {устунлар бўйича йиғиндиларни ҳисоблаймиз }
    j:=1;
repeat
    s:=0; { жорий устун элементлари йиғиндиси }
    for i:=1 to n do s:=s+a[i,j];
    j:=j+1;
    if s <> sum then ok:=FALSE;
until (not ok) or (j > n);
if ok then
    {бу ерда ҳар бир сатр элементлари йиғиндиси ҳар бир устун
    элементлари йиғиндиси ва бош диагонал элементлари
    йиғиндисига тенг }
begin
    {иккинчи бош диагонал элементлари йиғиндисини
    ҳисоблаймиз }

```

```

        s:=0;
        j:=n;
        for i: =1 to n do
            begin
                s:= s+a[i, j];
                j:= j - 1
            end;
        if temp <> sum then ok:=FALSE;
    end;
end;
write ('Киритилган матрица ');
if not ok
    then write ('афсункор квадрат бўлмайди ');
writeln ('афсункор квадрат бўлади. ');
readln;
end.

```

147 - масала

```

{ Универсиада ўйинлари натижаларини яқунлаш }
const
    n=14; { иштирокчи вилоятлар сони }
    viloyat: array [1..n] of string [9]=('Андижон','Бухоро', 'Жиззах',
        'Навоий','Наманган','Самарканд','Сурхондарё', 'Сирдарё',
        'Тошкент', 'Фарғона', 'Хоразм', 'ҚҚАССР' 'Қашқадарё',
        'Тошкент ш');
var
    { натижалар жадвали }
    natija: array [1.. n+1, 1..5] of integer;
    { n+1 - сатр жадвални саралашда буфер сифатида ишлатилади }
    i,j: integer;
    max: integer; { очколар сони максимал даражада саралашда
        ишлатиладиган жадвал сатри рақами }
    buf: string [9];
begin
    writeln ('Универсиада ўйинлари натижаси');

```

```

writeln ('Бир сатрдан олтин, кумуш ва бронза медаллари сонини
        киритинг. ');
{бошланғич маълумотларни киритиш }
for i:=1 to n do
    begin
        write (viloyat [i], ' → ');
        read (natija [i,1], natija [i,2]); {олтин ва кумуш медаллар сони}
        readln (natija [i,3]);          {бронза медаллар сони }
    end;
{умумий медаллар ва очколар сонини ҳисоблаймиз }
for i:=1 to n do
    begin
        natija [i, 4]:= natija [i,1]+ natija [i,2]+ natija [i,3];
        natija [i, 5]:= natija [i,1]*7+ natija [i,2]*6+ natija [i,3]*5;
    end;
{оддий танлаш усули билан очколар сонига мос равишда массивларни
саралаш}
for i:=1 to n-1 do
    begin
        {жадвал қисмидан i-сатрдан бошлаб, natija [j,5] элемент максимал
        бўлган j-сатрни топиш}
        max:=i; {бу i-рақамли сатр бўлсин}
        for j:=i+1 to n do
            if natija [j,5] > natija [max,5] then max:=j;
            {max рақамли сатрни i рақамли сатр билан алмаштириш.}
            {Буфер сифатида, жадвалнинг ишлатилмайдиган охириги сатри
            фойдаланилади.}
            buf:= viloyat [i];
            viloyat [i]:= viloyat [max];
            viloyat [max]:=buf;
        for j:=1 to 5 do
            begin
                natija [n+1,j]:= natija [i,j];
            end;
        for j:=1 to 5 do

```

```

begin
    natija [i,j]:= natija [max,j];
end;
for j:=1 to 5 do
begin
    natija [max,j]i:= natija [n+1,j];
end;
end;
{бу ерда жадвал тартибга солинган}
writeln;
writeln ('Талабалар Универсиадаси якунлари:');
writeln ('Вилоят': 12 , 'Олтин':8, 'Кумуш':8, 'Бронза':8, 'Жами': 8,
'Oчколар':8);
for i:=1 to n do
begin
write (i:2, viloyat [i]:10);
for j:=1 to 5 do
write (natija [i,j]:8);
writeln;
end;
readln;
end.

```

148 - масала

{«Сонни топ» ўйини - мураккаблаштирилган }

const

n=3 {мураккаблик даражаси – сондаги рақамлар миқдори}

var

o'yinchi: **array** [1..n] **of** char; (ўйинчи комбинацияси)

comp: **array** [1..n] **of** char; (компьютер комбинацияси)

a: **array** [1..n] **of** boolean; {агар компьютернинг i-рақамли ўйинчи рақамларидан биронтаси билан мос тушса, a [i] = TRUE)

top:integer; {сон топилди}

o'rin:integer; {улардан ўз ўрнида}

```

i,j: integer;           { массив индекслари }
begin
  writeln ('Компьютер уч хонали сонни ўйлади. ',
          'Сиз уни топишингиз керак, ');
  writeln ('Навбатдаги сон киритилгандан кейин Сизга нечта рақам');
  writeln ('топилганлиги улардан нечтаси ўз ўрнида эканлиги айтилади');
  writeln ('Сонлар киритилгач <Enter> ни босинг. ');
  {Компьютер ўз сонини «ўйлайди» }
  randomize;
  for i:=1 to n do
    comp [i]:= chr (random (10) +49); {48 - '0' белги коди }
  write ('Компьютер ўйлади: '); for i:=1 to n do
write (comp [i]); writeln;
  repeat
    write ('Сизнинг вариантингиз → ');
    {ўйинчи варианты олиш }
    for i:=1 to n-1 do
      read (o'yinchi [i]);
    readln (o'yinchi [n]);
    for i:=1 to n do a[i]:=FALSE;
    {a[i] = TRUE, агар компьютер сонининг i-чи рақами ўйинчи сони
      рақамларидан биттаси билан мос келса, нечта рақам
      топилганлиги текширилади }
    top:=0;
    for i:=1 to n do {ўйинчининг ҳар бир рақами текширилади }
      for j:=1 to n do
        begin
          if (o'yinchi [i] = comp[j]) and not a[j] then
            begin
              top:= top+1;
              a[j]:=TRUE; {компьютернинг бу рақамини ўйинчининг
                текширилмаган бошқа қолган рақамлари билан
                таққослаш ман этилади }
            end;
        end;
    end;
end;

```

```

    {улардан нечтаси ўз ўрнида эканлиги текширилади}
    o'rin:=0;
for i:=1 to n do
        if o'yinchi [i] = comp[i] then o'rin:= o'rin+1;
        writeln ('Топилди:', top, '. Ўз ўрнида:', o'rin);
until (top= n) and (o'rin= n);
    writeln ('***СОННИ ТОПДИНГИЗ!*** ');
    write ('Ишни тугатиш учун <Enter>ни босинг. ');
    readln;
end.

```

149 - масала

```

{ Телеграф – Морзе алифбоси ёрдамида ахборотни узатиш. }
{ Изоҳ: тўхтатишлар катталигини ўстириш мумкин. }
uses Crt;
const
    morse:  array [128 .. 159] of string [4] =(
        '-.', '-...', '-.---', '--.', {А,Б,В,Г}
        '-..', '!', '...-', '--..', {Д,Е,Ж,З}
        '.. ', '---', '-.-.', '-..', {И,Й,К,Л}
        '--', '-.', '---', '---.', {М,Н,О,П}
        '-.', '...', '-', '..-', {Р,С,Т,У}
        '..-', '...', '-.-.', '---.', {Ф,Х,Ц,Ч}
        '----', '---.', '-.-.', '-.-.', {Ш,Щ,Ъ,Ы}
        '-.-.', '-.-.', '-.-.', '-.-.', {Ь,Э,Ю,Я}
    );
    {узатиш параметрлари }
    TONE =100;  { сигнал частотаси (гц) }
    L1=50;      { «нуқта» узунлиги (мс) }
    L2=100;     { «тире» узунлиги (мс) }
    L3=50;      { нуқталар ва битта ҳарф тиреси ўртасида сукут-пауза (мс) }
    L4=100;     { ҳарфлар ўртасидаги сукут-пауза (мс) }
    L5=150;     { сўзлар ўртасидаги сукут-пауза (мс) }
var
    mes: string;  { ахборот }

```

```

sim: string [4]; {Морзеда символларни кодлаштириш нукта ва тирелар
                кетма-кетлигидан иборат}
belgi: string [1]; {"узатиладиган" белги - тире ёки нукта}
i,j: integer;      {символ ва белги рақами }

```

```
begin
```

```

  ClrScr;
  writeln ('*** Телеграф ***');
  writeln ('Узатилиши керак бўлган ахборотни киритинг');
  writeln (' (фақат бош рус ҳарфларидан фойдаланинг)');
  write ('→ ');
  readln (mes);
  for i:=1 to Length(mes) do

```

```
  begin
```

```
    if (mes[i] >= 'A') and (mes[i] <='Я') then
```

```
      begin
```

```

        {ахборотнинг навбатдаги ҳарф кодини аниқлаш (ф-я Ord) ва
        кодлар жадвалидан массивнинг тегишли элементи – нукта ва
        тирелар кетма-кетлигини ҳосил қилиш}

```

```
        sim:=morse [ord (mes[i])];
```

```
        j:=1;
```

```
        repeat
```

```
          if (sim[j] = '-') or (sim[j] = '.') then
```

```
            begin
```

```
              write (sim[j]);
```

```
              sound (1000);
```

```
              case sim [j] of
```

```
                ' . ': Delay (50);
```

```
                ' - ': Delay (100);
```

```
              end;
```

```
              NoSound;
```

```
              Delay (50);
```

```
            end;
```

```
            j:=j+1;
```

```
          until ((sim [j] = ' ') or (j>4));
```

```
          Delay (100);      {ҳарфлар ўртасида сукут-пауза}
```

```
        end
```

```

else
  if mes [i] = '' then {сўзлар ўртасида бўшлиқ}
  begin
    write ( ' '); {ахборот сўзлари ўртасида бўшлиқ}
    Delay (150);
  end;
end
writeln;
writeln ('Ахборот узатилди!');
writeln ('Дастур билан ишни тугатиш учун <Enter>ни босинг');
readln;
end.

```

3.1.10. ФУНКЦИЯЛАР ИШТИРОКИДАГИ ДАСТУРЛАР

150 - масала

{ max функция икки сондан бирига катта қийматни беради }

```
function max (a,b: integer): integer;
```

```
begin
```

```
  if a > b
```

```
    then max:=a
```

```
    else max:=b;
```

```
end;
```

151 - масала

{Қуйидаги функция сонларни таққослаш натижасини муносабат белгилари кўринишида беради }

```
function Compare(a,b: real): char;
```

```
begin
```

```
  if a > b then Compare:='>'
```

```
  else
```

```
    if a < b then Compare:='<'
```

```
    else Compare:='=';
```

```
end;
```

```
var
```

```
  x1,x2: real;      {таққосланадиган сонлар }
```

```
  nat: char;       {таққослаш натижаси }
```

begin

```
writeln ('Икки сонни киритинг ва <Enter>ни босинг');
write ('→');
readln (x1,x2);
nat:= Compare(x1,x2);    {дастурчи функциясини чақириш}
writeln (x1:6:2,nat,x2:6:2);
readln,
```

end.

152 - масала

{Электр занжири қаршилигини ҳисоблаш}

function Qar (r1,r2: real; t: integer): real;

{r1,r2 – қаршилик қийматлари, t – уланиш тури:

1 –кетма-кет;

2 - параллел.

Агар уланиш тури нотўғри кўрсатилган бўлса, 1 ни беради}

begin

if t=1 **then** Qar:= r1+r2;

if t=2 **then** Qar:= r1*r2/(r1+r2)

else Qar:=1; {уланиш тури хато кўрсатилган}

end;

153 - масала

{Логарифм хоссаларидан фойдаланиб, сон даражасини ҳисоблаш}

function InDar (a,b:real) :real;

begin

{A нинг B-чи даражаси C га тенг.

Тенгликнинг иккала томонини логарифмланиб қуйидаги ҳосил қиланади:

$$B * \ln(A) = \ln(C)$$

C қийматни топиш керак бўлгани учун E нинг $v * \ln(A)$ даражасини ҳисобланади. Бу ифоданинг қиймати C га тенг, шуни ҳисоблаш талаб қилинган эди.}

InDar:=exp (b*ln(a));

end;

```

var
  a: real; {сон}
  b: real; {даража}
  c: real; {даражадаги сон}
begin
  writeln ('Сонни ва даража кўрсаткичини киритинг');
  readln (a,b) ;
  c:= InDar (a,b) ;
  writeln (a:б:3, ' даражаси ',b:б:3,' = ',c:б:3);
  readln;
end.

```

154 - масала

```

{Омонатга кўра даромадни ҳисоблаш. }
function Daromad (sum: real; {омонат жами}
  omonat: real; {йиллик омонат фоизи}
  muddat: integer {омонат муддати (кун) }): real;
begin
  Daromad:= sum*(omonat/100/365)* muddat; {365 йилдаги кунлар
  сони}
end;

```

155 - масала

```

{Белги унли ҳарф бўлишини текшириш}
Function Unli (sim:char): boolean;
const
  {унли ҳарфлар }
  ListofUnli: string = 'АаЕеИиОоУуЭэЮюЯя Ўў';
var
  p: byte; {текширилаётган белгининг унлилар рўйхатидаги ўрни}
begin
  p:=O'rin (sim,ListOfUnli); { ! ! ! ! }
  if p <> 0 {белги рўйхатда топилди}
  then Unli:=True
  else Unli:=False;

```

```
end;
```

156 - масала

```
{Сатрдан бошланғич бўшлиқларни олиб ташлаш }
```

```
function LTrim (satr: string): string;
```

```
begin
```

```
while (Ord(' ', satr) = 1) and (length (satr) > 0) do
```

```
delete (satr,1,1);
```

```
LTrim: = satr;
```

```
end;
```

```
{LTrim функция ишини текшириш }
```

```
var
```

```
s:string [80]; {satr}
```

```
begin
```

```
writeln ('Сатрдан бошланғич бўшлиқларни олиб ташлайди. ');
```

```
write ('Сатрни киритинг → ');
```

```
readln (s);
```

```
write ('Бошланғич бўшлиқларсиз сатр:', LTrim(s));
```

```
readln;
```

```
end.
```

157 - масала

```
{Босма ҳарфларни ёзма ҳарфларга айлантириш }
```

```
{UpCase стандарт процедураси ўрнини босади }
```

```
function UpCase (satr:string): string;
```

```
var
```

```
i:integer;
```

```
begin
```

```
for i:=0 to Length (satr) do { белгилар нолдан бошлаб рақамланади! }
```

```
case satr [i] of
```

```
{ лотин ҳарфлари }
```

```
'a'..'z': UpCase[i]:=chr(ord(satr [i])-32);
```

```
{ рус ҳарфлари }
```

```
'а'..'п': UpCase[i]:=chr[ord(satr [i])-32];
```

```

        'p'..'я': UpCase[i]:=chr(ord(satr [i])-80);
    else    { қолган белгилар алмаштирилмайди }
        UpCase[i]:=st[i];
    end;
end;

{UpCase функциясини ишлатишга мисол}
var
    satr: string;
begin
    writeln ('Матнни киритинг ва <Enter>ни босинг');
    write ('→ ');
    readln (satr);
    writeln (UpCase (satr) );
    readln;
end.

```

158 - масала

```

    {Квадрат тенгламани ечиш}
function KvadrT (a,b,c: real; var x1,x2: real): integer;
    { a,b,c – тенглама коэффицентлари }
    { x1,x2 – тенглама илдизлари }
    { функция қиймати – илдизлар сони ёки, -1, агар бошланғич
    қийматлар хато бўлса, }
var
    d: real; { дискриминант }
begin
    if a = 0 then KvadrT:= -1
    else
        begin
            d:=b*b – 4*a*c;
            if d < 0 then
                KvadrT:= 0 {тенглама ечимга эга эмас}
            else begin
                if d > 0

```

```

        then KvadrT:=2      {иккита ҳар хил илдизлар }
        else KvadrT:=1;    {илдизлар бир хил }
        x1:= (-b+Sqrt(d))/(2*a);
        x2:= (-b-Sqrt(d))/(2*a);
    end;
end;
end,
{асосий дастур}
var
a,b,c: real;      {тенглама коэффициентлари }
x1,x2: real;     {тенглама илдизлари }

begin
writeln ('Квадрат тенгламани ечиш');
writeln ('Битта сатрда коэффициентларни киритинг ва <Enter> ни
        босинг');
write ('→');
readln (a,b,c) ;
case KvadrT (a,b,c,x1,x2) of
-1: writeln ('Бошланғич қийматлар хатоси. ');
0: writeln ('Тенглама ечимга эга эмас. ');
1: writeln ('x=',x1:6:2, ' Илдизлар бир хил. ');
2: writeln ('x1=',x1:6:2, ' x2=' ,x2: 6:2) ;
end;
readln;
end.

```

159 - масала

```

uses Crt;
var
a: integer;      {фойдаланувчи киритган сон }
{Функция GetInt битта ёки иккита рақамдан иборат бутун мусбат сонни
киритиш учун ишлатилади. Киритиш вақтида таҳрирлаш учун
<Backspace>ни ишлатиш мумкин.
<Enter> босилганда функция киритилган сонни қайтаради.}

```

```

function GetInt: integer;
const
    K_BACK=8;           { <Baskspace> клавиши коди }
    K_ENTER=13;        { <Enter> клавиши коди }
var
    ch:char;           { белги }
    dec:byte;          { белги коди }
    buf:array [1..2] of char;  { киритилган рақамлар }
    n: integer;        { киритилган рақамлар сони }
    x,y: integer;     { курсор ўрни }
begin
    buf [1]:=' ';    buf[2]:=' ';
    n: = 0;
    repeat
        ch:=Readkey;
        dec:=ord(ch) ;
        if (ch>= '0') and (ch<='9') and (n<2) then
            begin
                write (ch);
                n:=n+1;
                buf [n] r=ch;
            end
        else
            if (dec=K_BACK) and (n>0) then
                begin
                    n:= n-1;
                    x:= WhereX;
                    y:= WhereY;
                    GotoXY (x-1,y);
                    write ( '  ');
                    GotoXY(x-1,y) ;
                end;
            until (n>0) and (dec=K_ENTER);
            {киритилган сатрни сонга айлантирамиз}
            if n = 2

```

```

then GetInt:= (ord (buf [1]) – 48) *10 + ord (buf [2] ) – 48
else GetInt:= ord (buf[1]) – 48;
end;
begin
ClrScr;
writeln (' *** GetInt функция ишини намойиш этиш ***');
writeln;
writeln (' GetInt функция битта ёки иккита рақамдан иборат');
writeln ('бутун мусбат сонни киритишда ишлатилади. ');
writeln ('Киритиш пайтида таҳрирлаш учун <Backspace>дан
        фойдаланиш мумкин. ');
writeln (' <Enter> босилганда функция киритилган сонни қайтаради. ');
writeln;
writeln ('Сонни киритинг ва <Enter>ни босинг');
write ('→ ');
a:=GetInt;
writeln;
write ('Сиз ',a, ' сонини киритдингиз');
readln;
end.

```

3.1.11. ПРОЦЕДУРАЛИ ДАСТУРЛАР

170 - масала

{Юлдузчалардан иборат сатрни чиқариш}

```

procedure StarLine (len: integer);

```

```

  var

```

```

    i:integer;

```

```

  begin

```

```

    for i:=1 to len do

```

```

      write('*');

```

```

  end;

```

172 - масала

{Рамкани чизиш}

```

procedure Frame (e:integer; t:integer; w:integer; h:integer);

```

```

    {e,t –юқори чап бурчак координаталари}
    {w – рамка кенглиги, h – рамка баландлиги}
var
    x,y:integer; {чиқариладиган белги координаталари}
    i:integer;
    c1,c2,c3,c4,c5,c6:char; {рамкани чизадиган белгилар}
begin
    c1:=chr(218); {юқори чап бурчак белгиси}
    {chr белги функцияни чақиришда рақами кўрсатилган
    белгини қайтаради}
    c2:=chr(196); {горизонтал линейка белгиси}
    c3:=chr(191); {юқори ўнг бурчак белгиси}
    c4:=chr(179); {вертикал линейка белгиси}
    c5:=chr(192); {қуйи чап бурчак белгиси}
    c6:=chr(217); {қуйи ўнг бурчак белгиси}
    GOTOXY (1,t);
    write (c1);
    for i:=1 to w-2 do {рамка юқори чегаралари белгилари}
        write (c2);
    write (c3);
    y:=t+1
    x:1+w-1
    for i:=1 to h-2 do {чап ва ўнг чегаралар белгилари}
        begin
            GoToXY (1,y);
            write (c4)
            GoToXY (x,y);
            write(c4)
            y:=y+1
        end;
    GoToXY (1,y);
    write (c5)
    for i:=1 to w-2 do {қуйи чегара белгилари}
        write (c2);
    write (c6);

```

end,

3.1.12. ФАЙЛЛАРГА ОИД ДАСТУРЛАР

181 - масала

{A: дискда файл яратиш ва унга фойдаланувчи томонидан киритилган 5 та бутун сонни ёзиш }

var

f: text; { матн файли }
n: integer; { сон }
i: integer; { сон ҳисобчиси }

begin

writeln (' Файл яратиш');
writeln (' 5 та бутун сонни киритинг. ');
writeln ('Хар бир сон киритилгач, <Enter>ни босинг');
Assign (f,'a:\numbers.txt');
Rewrite(f); { қайта ёзиш режимида очинг }
for i:=1 **to** 5 **do**

begin

 write ('→');
 readln(n);
 writeln(f,n);

end;

 close(f); { файлни ёпинг }
 writeln ('Киритилган сонлар 'a:\numbers.txt файлида ёзилган');
 readln;

end.

182 - масала

{a:\numbers.txt файлига фойдаланувчи киритган 5 та бутун сонларни қўшимча ёзиш }

var

f: text; { матн файли }
n: integer; { сон }
i: integer; { сонлар ҳисобчиси }

begin

```

writeln (' a:\numbers.txt файлига қўшимча');
writeln ('Бешта бутун сон киритинг. ');
writeln ('Хар бир сон киритилгач, <Enter>ни босинг').
Assign (f, 'a:\numbers.txt');
Append(f);    { қўшиш режимида файлни очиш }
for i:=1 to 5 do
  begin
    write('→'),
    readln (n);
    writeln (f,n);
  end;
Close(f);      { файлни ёпиш }
writeln ('Киритилган сонлар a:\numbers.txt файлга қўшилган');
readln;
end.

```

183 - масала

```

{ a:\numbers.txt файлининг таркибини экранга чиқариш }
var
  f: text; { матн файли }
  n: integer; { сон }
begin
  writeln (' a:\numbers.txt файл таркиби');
  writeln ('-----')
  Assign (f, 'a:\numbers.txt');
  Reset (f);    { ўқиш учун файлни очиш }
  While not EOF (f) do { ҳозирча файл охирига етилмаган }
    begin
      readln (f,n);      { файлдан сонни ўқиш }
      writeln (n);      { ўқилган сонни экранга чиқариш }
    end;
  Close (f);      { файлни ёпиш }
  writeln ('-----')
  readln;

```

184 - масала

{a:\numbers.txt файлидаги сонларнинг ўрта арифметигини ҳисоблаш}

var

```
f: text;      { матн файли }
n: integer;   { файлдан ўқилган сон }
miq: integer; { ўқилган сонлар миқдори }
sum: integer; { ўқилган сонлар жами }
o'a: real;    { ўрта арифметик }
```

begin

```
writeln (' a:\numbers.txt файлидаги сонларнинг, ');
writeln ('ўрта арифметигини ҳисоблаш');
writeln ('Файлдан ўқиш. Кутинг. ');
sum:=0;
miq:=0;
Assign (f, 'a:\numbers.txt');
Reset (f);   { файлни ўқиш учун очиш }
While not EOF (f) do { ҳозирча файл охирига етилмаган }
```

begin

```
  readln (f,n);      { файлдан сонни ўқиш }
  sum:=sum+n;
  miq:= miq +1;
```

end;

```
Close (f);          { файлни ёпиш }
o'a:=sum/ miq;
writeln ('Ўқилган сонлар: ', miq);
writeln ('Сонлар йиғиндиси: ',sum);
writeln ('Ўрта арифметик: ',o'a:9:2);
readln;
```

end.

185 - масала

{Исми фойдаланувчи томонидан кўрсатилган файл таркибини экранга чиқариш }

uses Crt;

var

```

f: text;                { матн файли }
fname: string [80];    { файл исми }
st: string;          { файлдан ўқилган сатр }
sts: integer; '        { экранга чиқарилган сатрлар сони }
key: char;              { фойдаланувчи босган клавиш }

```

begin

```

ClrScr;    { экранни тозалаш }
writeln ('Матн файлини кўздан кечириш');
writeln ('Файлнинг тўла исмини киритинг ва <Enter>ни босинг');
write ('→ ');
readln (fname);
Assign (f,fname);
Reset(f);  { файлни ўқиш учун очиш }
ClrScr;
sts:=0;
While not EOF(f) do { ҳозирча файл охирига етилмаган }

```

begin

```

    readln (f,st);    { файлдан сонни ўқиш }
    writeln (st);
    sts:=sts+1;
    if sts = 23 then { навбатдаги 23 та сатр чиқарилди }

```

begin

```

    writeln;
    write ('Киритишни давом эттириш учун ихтиёрий
           клавишни босинг...');
    key:= Readkey;
    GotoXY (1,WhereY); { курсорни жорий сатр бошига этиш }
    DelLine; { «Давом эттириш учун...» ахборотини
              йўқотиш }
    sts:=0;

```

end;

end;

```

Close (f);    { файлни ёпинг }
writeln;
write ('Кўришни давом эттириш учун ихтиёрий клавишни босинг...');

```

```
key:=Readkey;
```

```
end.
```

186 - масала

{a:\phone.txt файлига исм, шариф ва телефон рақамини қўшимча ёзиш.
Агар файл дискда бўлмаса, уни яратиш }

```
label
```

```
1;
```

```
var
```

```
f: text;           { матн файли }
```

```
fam: string [15]; { фамилия }
```

```
name: string [15]; { исм }
```

```
tel: string [9];   { телефон рақами }
```

```
begin
```

```
writeln ('Телефон маълумотномасига қўшимча');
```

```
Assign (f, 'a:\phone.txt');
```

```
{ $I }
```

```
Append (f);   { аввал қўшиш режимида очамиз }
```

```
if IOResult <> 0 then
```

```
    { файл phone.txt A: дискда бўлмаслик эҳтимоли бор, }
```

```
    { уни яратамиз }
```

```
    begin
```

```
        Rewrite (f);   { қайта ёзиш режимида очиш }
```

```
        if IOResult <> 0 then
```

```
            begin
```

```
                writeln (' A: дискка мурожаат этишда хатолик');
```

```
            goto 1;
```

```
            end;
```

```
    end;
```

```
    { фойдаланувчидан маълумотлар оламиз }
```

```
    write ('Фамилия →');
```

```
    readln (fam);
```

```
    write ('Исм → ');
```

```
    readln (name);
```

```
    write ('Телефон →')
```

```

readln (tel);
{ ва уларни файлга ёзамиз }
writeln (f,fam) ;
writeln (f, name);
writeln (f,tel);
close (f);
writeln ('Ахборот кўшилди. ');
1: writeln ('Ишни тугатиш учун <Enter>ни босинг. ');
readln;

```

end.

187 - масала

{ Телефон маълумотномасидан излаш }

label 2;

var

```

f: text;           { матн файли }
nam: string [15]; { излаш учун фамилия }
n: integer;       { сўровни қаноатлантирадиган ёзувлар миқдори }
fam: string [15]; { фамилия }
name: string [15]; { исм }
tel: string [9];  { телефон рақами }

```

begin

```
writeln ('Телефон маълумотномасидан излаш');
```

```
Assign (f,'a:\phone.txt');
```

```
{ $I- }
```

```
reset (f);      { ўқиш учун файлни очамиз }
```

```
if IOResult <> 0 then
```

```
  { phone.txt файли А: дискда бўлмаслик эҳтимоли бор }
```

```
  begin
```

```
    writeln (' a:\phone.txt файлидан фойдаланиб бўлмайди');
```

```
    goto 2;
```

```
  end;
```

```
repeat
```

```
  write ('Фамилия → ');
```

```
  readln (nam);
```

```

if Length (nam) <> 0 then { фойдаланувчи сатрни киритди }
    begin
        reset(f); { файлни бошидан кўриб чиқамиз }
        n:= 0;
        while not EOF(f) do { файлнинг ҳаммасини кўриб чиқамиз }
            begin
                { файлдан ўқиймиз }
                readln (f,fam);
                readln (f,name);
                readln (f,tel);
                if fam = nam then { керакли фамилия топилди }
                    begin
                        writeln (fam,' ',name,' ',tel);
                        n:= n+1;
                    end;
            end;
        end;
        if n = 0 then writeln (nam, 'тўғрисида ахборотлар йўқ');
        end;
    until Length (nam) = 0;
    2:writeln ('Ишни тугатиш учун <Enter>ни босинг. ');
    readln;
end.

```

188 - масала

{Тестдан ўтказишнинг универсал дастури }

uses Crt;

label

 nomer;

var

 fname: **string** [40];

 {тестдаги файл исми }

 f: text;

 {тест файли }

 JamiSav: integer;

 {тест саволлари миқдори }

 To'g'Jav: integer;

 { жорий савол учун тўғри жавоблар миқдори }

 nJav: integer;

 { мазмуни бир хил бўлган жавоблар сони }

 To'g': integer;

 {тўғри жавоб рақами }

```

Jav: integer;           { фойдаланувчи танлаган жавоб рақами }
sat: string;          { тест файлидан ўқиладиган сатр }
p: integer;            { тўғри жавоблар фоизи }
i: integer;           { цикллар ҳисобчиси }

```

```
begin
```

```
if ParamCount = 0 then
```

```
begin
```

```
    writeln ('Тест саволлари файли берилмаган!');
```

```
    writeln ('Буйруқ сатри: test ТестФайлиИсми');
```

```
    goto nomer;
```

```
end;
```

```
fname: = ParamStr (1); { буйруқ сатридаги файл исми }
```

```
Assign (f,fname);
```

```
{ $I- }           { бу изоҳ эмас, балки компиляторга буйруқ }
```

```
Reset(f);        { ўқиш учун файлни очиш }
```

```
{ $I+ }
```

```
if IOResult <> 0 then
```

```
begin
```

```
    writeln (fname, 'тест файли топилмади');
```

```
    goto nomer;
```

```
end;
```

```
writeln ('Ҳозир Сизга тест таклиф этилади.');
```

```
writeln ('Ҳар бир саволга жавобларнинг бир нечта варианты берилади);
```

```
writeln ('Сиз тўғри жавоб рақамини киритишингиз ва <Enter>  
клавишини босишингиз керак');
```

```
writeln;
```

```
writeln ('Муваффақият! Тестни бошлашдан олдин <Enter>ни босинг');
```

```
TextBackGround(Blue);
```

```
ClrScr;
```

```
JamiSav:=0 ;
```

```
To'g':=0;
```

```
while not EOF(f) do
```

```
begin
```

```
    {ClrScr;}
```

```
    JamiSav:= JamiSav + 1;
```

```

Readln (f,sat);           {файлдан саволни ўқиймиз}
TextColor(White);
writeln(sat);           {Экранга саволни чиқарамиз}
readln (f,nJav, To'g'); {моҳияти бир хил бўлган жавоблар сонини
                        ва тўғри жавоб рақамини ўқиймиз.}
TextColor (LightGray);
for i:=1 to nJav do {ўқиймиз ва моҳияти бир хил жавобларни
                        чиқарамиз }

    begin
        readln (f,sat);
        writeln(i,'.',sat);
    end;
    writeln;
    write ('Сизнинг танловингиз →');
    readln(Jav);
    if Jav = To'g' then To'g'Jav:= To'g'Jav+ 1;
    writeln;
end;
{Тест натижаларини қайта ишлаш }
{тўғри жавоблар фоизини ҳисоблаш }
p:= Round ((To'g'Jav/ JamiSav)*100);
write ('Сизнинг баҳонгиз ');
case p of
    100:writeln ('АЪЛО!');
    80..99:writeln ('ЯХШИ. ');
    60..79:writeln ('ҚОНИҚАРЛИ. ');
    else writeln ('ЁМОН!');
end;
nomer: writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
readln;
end.

```

189 - масала

```

{Дюймлардан миллиметрларга ўтиш жадвалини экранга, принтерга ёки
  файлга чиқариш }

```

var

```
f:text;           { чиқариш файли }
fname:string;    { чиқариш файли исми }
dest:integer;    { 1 - экранга, 2 - принтерга, 3 - файлга }
d: real;         { дюймлардаги миқдор }
m: real;         { миллиметрлардаги миқдор }
```

begin

```
writeln ('*** Дюймлардан миллиметрларга ўтиш жадвали ***');
writeln ('Натижани чиқаради:');
writeln ('1 - экранга;');
writeln ('2 - принтерга; ');
writeln ('3 - файлга. ');
writeln ('1 дан 3 гача сонларни киритинг ва <Enter>ни босинг');
write ('Сизнинг танловингиз → ');
readln (dest);
case dest of
  1: fname:= ' ';           { экранга }
  2: begin                 { принтерга }
      fname:='prn',;
      write ('Принтерни ёқинг ва <Enter>ни босинг');
      readln;
    end;
  3: begin                 { файлга }
      write ('Чиқариш учун файл исмини беринг → ');
      readln (fname);
    end;
end;
assign (f,fname);
rewrite (f);
writeln (f, ' -----' );
writeln (f, ' Дюймлар Миллиметрлар');
writeln (f, ' -----' );
d:= 0.5;
while d < 10 do
  begin
```

```

        m:=25.4*d;          { 1 дюйм - 25,4 мм }
        writeln (f,d:6:1, m:10:1);
        d:= d+0.5;
    end;
writeln (f,'-----');
close (f);
if dest = 3
    then writeln ('Жадвал' ,fname, 'файлида ёзилган');
    writeln ('Дастур ишини тугатиш учун <Enter>ни босинг');
    readln,
end.

```

3.1.13. РЕКУРСИЯЛИ ДАСТУРЛАР

192 - масала

{«Факториал» рекурсив функцияси)

```
function factorial (k:integer):integer;
```

```
begin
```

```
    if k = 1
```

```
        then factorial:= 1
```

```
        else factorial:= k*factorial (k-1);
```

```
    end;
```

```
var
```

```
    n: integer;      { факториали ҳисобланиши керак бўлган сон }
```

```
    f: integer;      { n сонининг факториали }
```

```
begin
```

```
    writeln (' Факториални ҳисоблаш,');
```

```
    writeln ('Факториали ҳисобланиши керак бўлган сонни киритинг');
```

```
    write ('→ ');
```

```
    readln (n);
```

```
    f:= factorial (n);
```

```
    writeln (n,'нинг факториали ', f, 'га тенг');
```

```
    readln;
```

```
end.
```

193 - масала

```
{Экранга безаклар чиқаради }
uses Graph, Crt;
{Безак элементини чизади}
procedure Elem (x,y, r,p: integer);
    {x, y, r – безак асосий элементи марказининг координаталари
    ва радиуси – безак тартиби }
begin
    if p >= 0 then
        begin
            Circle (x, y, r) ;      {Радиуси r га тенг, координата маркази
                                   x,y бўлган айлана чизади}
            Delay (100) ;          {Дастур бажарилишини 100 милли
                                   секунд тўхтатиб туради}
            Elem (x+r, y, Round (r/2), p-1);
            Elem (x, y-r, Round (r/2), p-1);
            Elem (x-r, y, Round (r/2), p-1);
            Elem [x, y+r, Round (r/2), p-1);
        end;
    end;
var
    grDriver:integer;      {драйвер}
    grMode:integer;       {график режим}
    grPath:string;        {драйвер файлига йўл}
    ErrCode:integer;      { график режим хатолик коди}
begin
    grDriver:=VGA;        {драйвер}
    grMode:=VGAHi;       {график режим}
    grPath:= 'e:\tp\bgi'; {драйвер жойлашган адрес}
    InitGraph (grDriver, grMode, grPath); {График режимни инициализация қилади. Параметрлар видеосистема драйверини, иш режимини, адресни билдиради}
    ErrCode:= GraphResult; {GraphResult – охириги бажарилган график операция натижасини қайтарувчи график режим функцияси}
```

```

if ErrCode <> grOK then
  begin
    writeln (' График режим хатосини инициализациялаш.')
    writeln ('Ишни тугатиш учун <Enter>ни босинг');
    readln;
    Halt (1);
  end;
  {Асосий дастур }
  Elem (320, 240, 60, 3);      { 3-тартибли безак чизамиз )
  OutText ('Дастур ишини тугатиш учун <Enter>ни босинг');
  readln;
end.

```

194 - масала

{Рекурсия механизидан фойдаланиб, n-бўғинли электр занжир қаршилигини ҳисоблаш }

var

```

r1,r2,r3: real;   { занжир ташкил топган қаршилиқлар катталиги }
n: integer;      { занжир бўғинлари сони (тартиби) }
rz: real;        { занжир қаршилиги }
{ n-чи тартибли занжир қаршилиги қиймати }

```

function Zanj (n: integer): real;

begin

if n= 1

then Zanj:=R1+R2+R3

else begin

rz: = Zanj (n-1);

Zanj:=R2* rz /(R2+ rz) ;

end;

end;

{асосий процедура }

begin

writeln ('Электр занжир қаршилигини ҳисоблаш.');

writeln ('Қаршилиқ катталиқларини киритинг (Ом) : ') ;

write ('r1 →');

```

readln (r1) ;
write ('r2 →');
readln (r2);
write ('r3 →');
readln (r3) ;
write ( 'Занжир тартиби →');
readln (n). ;
writeln ('Занжир қаршилиги: ',Zanj(n) :6:2, ' Ом');
{қаршилиқлар катталиги Zanj процедурасига r1, r2 ва r3 глобал
ўзгарувчилар орқали узатилади }
readln;
end.

```

195 - масала

```

{Мураккаб электр занжир схемасини чизади.}
{Занжир схемасини чизишда рекурсиядан фойдаланилади.}
Uses Graph;
const
  {тур қадами }
  dx = 7;
  dy = 7;
var
  grDriver:integer;
  grMode:integer;
  grPath:string;
  ErrCode:integer;
  x,y: integer;
  k: integer;      {занжир тартиби }
  { k-чи тартибли занжир схемасини чиқаради }
Procedure Zanj (k: integer; x,y: integer);
begin
  SetColor (Green);
  Line (x, y, x+2*dx, y);
  Rectangle (x+2*dx, y-dy, x+6*dx, y+dy) ;

```

```

Line (x+6*dx, y, x+8*dx, y);
OuttextXY (x+3*dx, y-3*dy,'R1');
SetColor (Yellow);
Line (x+8*dx, y, x+8*dx, y+2*dy);
Rectangle (x+7*dx, y+2*dy, x+9*dx, y+6*dy);
Line (x+8*dx, y+6*dy, x+8*dx, y+8*dy);
OuttextXY (x+10*dx, y+2*dy, 'R2');
SetColor (LightGray) ;
Line (x, y+8*dy, x+2*dx, y+8*dy);
Rectangle (x+2*dx, y+7*dy, x+6*dx, y+9*dy);
Line (x+6*dx, y+8*dy, x+8* dx, y+8*dy);
OuttextXY (x+3*dx, y+5*dy, 'R3');
if k>1 then Сеп(k-1,x+8*dx,y);

```

```
end;
```

```
begin
```

```

grDriver:=VGA;           { драйвер }
grMode:=VGAHi;          { график режим }
grPath: ='e:\tp\bgi';   { драйвер файлига йўл }
InitGraph (grDriver, grMode, grPath); { график режимни
                               инициализациялаш, параметрлар видеосистема
                               драйверини, иш режимини, адресни билдиради }
ErrCode: = GraphResult; { охирги бажаррилган
                          график операция натижасини кайтарувчи график
                          режим функцияси }

```

```
if ErrCode <> grOK then
```

```
begin
```

```

writeln ('График режим хатосининг инициализацияси .');
writeln ('Ишни тугатиш учун <Enter>ни босинг');
readln;
Halt (1)

```

```
end;
```

```
OutTextXY (10,10, 'Занжир тартибини киритинг ва <Enter>ни  
            босинг');  
readln (k);  
Zanj (k,10,50);  
readln;  
CloseGraph;  
end.
```

§ 3.2. ЮҚОРИ МУРАККАБЛИКДАГИ МАСАЛАЛАР АЛГОРИТМЛАРИ ВА ДАСТУРЛАРИ

Бу бўлимда дастурдан аввал масала ечимининг алгоритмлари келтирилади. Бу баъзан, фақат алгоритмик тилда ёзилиши қолган, тугалланган фикр, баъзи ҳолларда эса ечим мулоҳазаси, бошқа ҳолларда эса дастурни тушунтирувчи изоҳ бўлади. Лекин ҳамма ҳолларда ҳам дастурни ўқиш енгиллаштирилган, ўзлаштириш керак бўлган дастур усуллари баён этилади. Амаллар аниқлиги, одатда, доимий кўпайтувчи аниқлигида кўрсатилган.

201 – масала

Алгоритм. «Кўндаланг диагонал»

Иккита сон аниқлаймиз: $P = \max(1, 1-k)$ ва $g = \min(n, m-k)$. Шунда изланаётган миқдор, ҳамма $j = p, p+1, \dots, g$ лар бўйича $A[k+j, j]$, элементлар йиғиндисига тенг бўлади (агар j нинг бундай қийматлари бўлмаса, яъни $p > g$ бўлса, у нолга тенг). A массивнинг ҳамма $m \times n$ элементлари қараладиган ечимни қониқарли эмас, деб қараши керак.

Дастур

```
Program Ko'ndalang_diagonal;  
const NN=10;  
      MM=10;  
var  
      m,n, j,k,p,g,s :integer;  
      A : array [1..MM,1..NN] of integer;  
begin  
      writeln ('M,N,K:=');  
      readln (m,n,k);  
      for p:=1 to m do  
          for j:=1 to n do  
              begin  
                  writeln ('A [' ,P, ', ',j, ']:=');  
                  readln (A[p,j]);  
              end;  
      if k>0 then p:=1 else p:=1-k;
```

```

if k+n<m then g:=n
      else
          g:=m-k;
          s:=0;
for j:=p to g do s:=s+A[k+j,j];
writeln(s);

```

end.

202 – масала

Алгоритм. «Квадратчалар».

Квадратчанинг бурчакларидаги ҳамма сонлар ҳар хил бўлса, уларнинг йиғиндиси $0+1+5+1=17$. Лекин бунинг акси ҳам тўғри эканлигини таъкидлаш мумкин. Бу дастурни соддалаштиради.

Дастур

Program Kvadratchalar;

const MM=50;

var

i,j,m,S :integer;

A : **array** [1..MM,1..MM] **of** integer;

begin

writeln ('M:=');

readln (m);

for i:=1 **to** m **do**

for j:=1 **to** m **do**

begin

 writeln ('A ['i','j']:=');

 readln (A[i,j]);

end;

 S:=0;

for i:=1 **to** m-1 **do**

for j:=1 **to** m-1 **do**

if (A[i,j]+ A[i,j+1]+ A[i+1,j]+ A[i+1,j+1]=R) **then** S:=S+1;

 writeln (S);

end.

203 – масала

Алгоритм. «Санок системалари»

Берилган сон миқдорини Горнер схемаси бўйича топиш керак:

$$N = ((\dots(M[9]*i + M[8])*i + \dots + M[2])*i + M[1])$$

N , бутун сонлар учун, мумкин бўлган максимал миқдордан ошмайди, деб фараз қилинади. N сонлар разрядлари (хоналари) j -системада N ни j га бутун бўлишдаги қолдиқлар сифатида ҳосил қилинади:

$$k = N/j : M = N - K*j : N = K \text{ ва ҳоказо.}$$

M нинг қолдиқларини масала шартига кўра, эслаб қолиш ва чоп этиш шарт эмас.

Дастур

```
Program Sanoq_sistemalar;
```

```
var
```

```
  i, j, k : integer;
```

```
  n : real;
```

```
  m : array [1..9] of integer;
```

```
begin
```

```
  writeln ('i,j:=');
```

```
  readln (i,j);
```

```
  for k:=9 downto 1 do
```

```
  begin
```

```
    write ('M[',k,']:=');
```

```
    readln (m[k]);
```

```
  end;
```

```
  writeln;
```

```
  n:=0
```

```
  for k:=9 downto 1 do n :=n*i+M[k];
```

```
  writeln (trunc(n));
```

```
  repeat
```

```
    write (trunc (n), mod j, ' ');
```

```
    n:= trunc(n) div j;
```

```
  until n=0;
```

```
  writeln
```

```
end.
```

204 – масала

Алгоритм «Календар» .

Кабиса бўлмаган йил ойларида кунларнинг сон бўйича M[1:11] жадвалини ҳосил қилиш қулай. «m» ни «n»га бўлишдаги қолдиқни аниқлаб берувчи MOD(m,n) ёки унга ўхшаш функцияни ҳам ишлатиш мумкин.

Дастур

Program Kalendar;

var

a,b,c,i,j : integer;

M : array [1..11] of integer;

function D (x:integer) : boolean;

begin

D:=(c mod x)=0;

end;

begin

writeln ('a,b,c:=');

readln (a,b,c);

writeln;

for i:=1 to 11 **do**

case i **of**

1,3,5,7,8,10: M[i]:=31;

4,6,9,11: M[i]:=30;

2: M[i]:=28;

end;

j:=a;

for i:=1 to b-1 **do** j:=j+m[i];

if (b>2) **and** (D(4) **and** **not** D(100) **or** D(400)) **then** j:=j+1;

writeln (j);

end.

205 – масала

Алгоритм «Сўзнинг матнга киритилиши».

Дастурда осон ўқилади. Изоҳга ҳожат йўқ.

Program So'zni_matnga_kiritish

const nn=100; kk=50;

var

i,n,j,k : integer;

ff : boolean;

x : **array** [1..nn] **of** integer;

y : **array** [1..kk] **of** integer;

begin

writeln ('k,n=');

readln(k,n);

writeln;

writeln ('массив x:');

for j:=1 **to** n **do** readln(x[i]);

writeln;

writeln ('массив y:');

for j:=1 **to** k **do** readln(y[j]);

writeln;

i:=0;

ff:=true;

while (i<=n-k) **and** ff **do**

begin

j:=1;

while j<=k **and** ff **do**

if x[i+j]<>y[j] **then** ff:=false

else j:=j+1;

if ff **then**

begin

writeln ('xa', i+1);

ff:=false;

end

else

begin

ff:= true;

i:=i+1;

```
        end;  
    end;  
    if ff then writeln ('yoq');  
end.
```

206 – масала

Алгоритм «Ноллар серияси»

Дастур оддий. Изоҳга ҳожат йўқ.

Дастур

```
Program Nollar_seriyasi  
const nn=100;  
var  
    i,n,t,max    : integer;  
    a : array [1..nn] of real;  
begin  
    writeln ('n=');  
    readln (n);  
    for i:=1 to n do readln(a[i]);  
    t:=0; max:=0;  
    for i:=1 to n do  
        begin  
            if a[i]<>0 then  
                begin  
                    if (max<t) then max:=t;  
                    t:=0;  
                end;  
            else t:=t+1;  
        end;  
    if max< then max:=t;  
    writeln;  
    writeln (max);  
end.
```

207 – масала

Алгоритм «Эгар нуқта»

Агар m_i i -сатрдаги a_{ij} элементларнинг энг кичик қиймати, M_j эса j устундаги a_{ij} элементларнинг энг катта қиймати бўлса,

$m_i \leq a_{ij} \leq M_j$, бўлади.

Бу бир нечта мулоҳазалар қилишга имкон беради:

а) ихтиёрий сатрнинг минимуми ихтиёрий устун максимумидан катта эмас, яъни ҳамма вақт $m_i \leq M_j$ шарт ўринли;

б) агар қандайдир i ва j учун $m_i = M_j$ тенглик бажарилса, минимумларнинг энг каттаси максимумларнинг энг кичиги билан мос келади, i сатрнинг j устун билан кесишишида эса эгар нуқта:

$m_i = a_{ij} = M_j$ ётади; (*)

в) агар a_{ij} -эгар нуқта бўлса, унинг учун (*) шарт бажарилади ва, демак, минимумларнинг максимуми, максимумларнинг минимумига тенг бўлади.

Айтганларни дастурга жорий этиш керак. Ҳар бир сатрда минимум m_i изланади, бу минимумлар ичидан максимум - M_a танланади ва у жойлашган i_0 сатр эслаб қолинади. Агар минимум « M_a » нинг жорий қийматидан кичик бўлиб қолса, сатрда минимумни излашни тўхтатиш керак.

Кейин, максимал элемент топилган « M_a » қийматга тенг устун изланади, агар бундай устун бўлмаса, эгар нуқта ҳам йўқ. Амалда дастурда, агар устунда M_a элементдан катта элемент бўлмаса а) пунктга кўра, устун, изланаётган устун эканлиги кўрсатилади.

Агар дастурда бу ерда келтирилган мулоҳазалардан фойдаланилмаса, у анча секин ишлайдиган бўлади.

Дастур

```
Program Egar_nuqta
const nn=20; mm=20;
label 1,2;
var
    i,n,j,m,io,mi,ma :integer;
    a : array [1..mm,1..nn] of integer;
begin
    writeln ('m,n=');
```

```

readln (m,n);
for i:=1 to m do
  begin
    writeln ('a[',i,',',j,']=');
    readln (a[i,j]);
  end;
for i:=1 to m do
  begin
    if (i>1) and (a[i,j] <=ma) then goto 1;
    if (j=1) or (a[i,j] <mi) then mi:=a[i,j];
  1 : end;
  ma:=mi; io:=i;
  for i:=1 to n do
    begin
      for i=1 to m do if a[i,j]> ma goto 2;
      writeln (io:3,j:3); exit;
    2: end;
  writeln(0);
end.

```

208 – масала

Алгоритм «Касрни қисқартириш».

Евклид алгоритмини дастурлаш мумкин. Унга кўра m ва n сонларнинг энг катта умумий бўлувчисини топиш ва унга қисқартириш керак.

Евклид алгоритми моҳиятини тушунтирамиз, $m_1 \geq m_2$ бўлсин, (m_1, m_2) жуфтликнинг ихтиёрий умумий бўлувчиси $(m_2, m_1 - m_2)$ жуфтликнинг, демак, (m_2, m_3) жуфтликнинг ҳам умумий бўлувчиси бўлади, бу ерда

$$m_3 = m_1 - (m_1 / m_2) * m_2$$

m_1 ни m_2 га бўлишдаги қолдиқ бўлади, шунинг учун $m_3 < m_2$ олдиндан маълум. Акси ҳам тўғри: (m_2, m_3) жуфтликнинг ҳар қандай умумий бўлувчиси (у.б.) (m_1, m_2) жуфтликнинг ҳам умумий булувчиси бўлади. Шунинг учун

$$\text{у.б. } (m_1, m_2) = \text{у.б. } (m_2, m_3)$$

у.б. (m_1, m_2) функцияда аргументни уни кичигига бўлгандаги қолдиғи билан кетма-кет алмаштира бориб,

$$\text{у.б. } (m_1, m_2) = \text{у.б. } (m_2, m_3) = \dots = \text{у.б. } (m_k, 0) = m_k$$

кетма-кетлик ҳосил қилинади, унда $m_1 \geq m_2 > \dots > m_k > 0$ ва m_k бошланғич m_1, m_2 сонларнинг энг катта умумий бўлувчиси бўлади.

Агар n катта сон бўлмаса, j -энг кичик қийматли $i/j = m/n$ касрни танлаб олиш мумкин. У, албатта, қисқармайди. Фақат касрлар тенглигини нисбатлар тенглиги билан эмас, кўпайтмалар тенглиги билан текшириш керак:

$$i * n = j * m$$

Бу таклиф этилган дастурга олиб келади.

Дастур

```
Program Kasrni_qisqartirish;
```

```
label 1;
```

```
var
```

```
    i,n,j,m :integer;
```

```
begin
```

```
    readln (m,n);
```

```
    writeln ('m=',m,'n=',n);
```

```
    for j:=1 to n do
```

```
        begin
```

```
            i:=j*m div n;
```

```
            if i*n= j*m then goto 1;
```

```
        end;
```

```
    1: writeln ('m/n',i,'/',j);
```

```
end.
```

209 – масала

Алгоритм «Массивларнинг қўшилиши»

Бу муҳим масала $m+n$ амалда бажарилиши керак. А ва В дан биринчи элементларини оламыз, улардан кичигини С массивга ёзамиз ва уни ўз массивдан навбатдагиси билан алмаштирамиз. Яна иккитадан кичигини танлаб, кейин С га киритамиз ва ҳоказо, ҳар бир таққослашдан кейин С га элемент қўшилади, демак, таққослаш $m+n$ дан кичик бўлади. Фақат

дастурнинг, массивлардан биттаси томом бўлганидан кейин ҳам, тўғри ишлашни таъминлаш керак.

Дастур

Program Massivlar_qo'shilishi;

const MM=100;

 NN=100;

 MN=200;

var

 m,n,i,j,k :integer;

 A : **array** [1..MM] **of** integer;

 B : **array** [1..NN] **of** integer;

 C : **array** [1..MN] **of** integer;

begin

 writeln ('M,N:=');

 readln (m,n);

 writeln ('**array** A:=');

for i:=1 **to** m **do** readln(A[i]);

 writeln ('**array** B:=');

for j:=1 **to** n **do** readln(B[j]);

 i:=1; j:=1;

for k:=1 **to** m+n **do**

begin

if ((i>m) **or** (A[i]>B[j])) **and** **not** (j>n) **then**

begin

 c[k]:=B[j]; j:=j+1;

end;

else

begin

 c[k]:=A[i]; i:=i+1;

end;

end;

 writeln ('**array** C:=');

for k:=1 **to** m+n **do** writeln (c[k]);

end.

210 – масала

Алгоритм «Мода»

Бошланғич массивни тартибга келтириш, шундан кейин бир қарашда қийматлар частотасини – энг кўп учрайдиган қийматни ҳисоблаш керак. Частотани ёзиш учун массив ташкил этишга ҳожат йўқ-тез учрайдиганлар номзодини эслаб қолиш ва унинг частотасини қаралган элементнинг топилган частотаси билан таққослаш етарли.

Агар қандайдир ҳолга кўра, тартиблаштиришни бажармаслик керак бўлса, қуйидагича йўл тугилади: навбатдаги $a[i]$ (бошида бу $a[1]$ элемент) ундан кейин келадиган элемент билан таққосланади. Бунда, биринчидан, $a[i]$ қийматнинг пайдо бўлиш частотаси ҳисобланади, иккинчидан, $a[i]$ га тенг бўлган $a[j]$ элементлари кетма-кет $a[i+1]$, $a[i+2]$... элементлари билан алмаштирилади. А массив охиригача қараб чиқилиб, $a[i]$ элемент частотаси аниқланганидан кейин $a[i+k]$ элемент частотаси топилади, $a[i]$ ва $a[i+k]$ ўртасидаги элементлар ўтказиб юборилади. Бу алгоритм самарали кўринади ва агар массивда ҳар хил элементлар бўлса, $n*n$ та амал бажарилиш керак бўлади. Бу фикрлар таклиф этилган дастурда амалга оширилган.

Дастур

```
Program Moda;  
const NN=100;  
var  
    n,m,am,i, j,k : integer;  
    A : array [1..NN] of integer;  
begin  
    writeln ('n:=');  
    readln (n);  
    for i:=1 to n do readln(a[i]);  
    writeln;  
    m:=0; i:=1;  
    while i+m<=n do  
    begin  
        k:=1;  
        for j:=i+1 to n do
```

```

        if a[j]=a[i] then
            begin
                a[j]:=a[i+k]; k:=k+1;
            end;
        if m<k then
            begin
                am:=a[i]; m:=k ;
            end;
        i:=i+k;
    end;
    writeln (am);
end.

```

211 – масала

Алгоритм «Марказий қишлоқ»

Бу масалани куйидагича баён этамиз: А массивнинг ҳар бир i сатрида $a[i,j]$ ($j \neq i$) сонлари ичидан энг каттаси танланади ва у $a[i,j]$ га қўшилади. Тегишли йиғиндини энг кичик қилувчи i топилиши керак. Масаланинг бундай қўйилишида унинг стандарт ечимга эгалигига йўл қўйилади.

Дастур

```

Program Markaziy_qishloq;
const KK=20;
var
    i,j,k,il,s,t :integer;
    A : array [1..KK,1..KK] of real;
begin
    writeln ('k:=');
    readln (k);
    for i:=1 to k do
        for j:=1 to k do
            begin
                writeln('a[',i,',',j,']=');
                readln(a[i,j]);
            end;
    for i:=1 to k do

```

```

begin
  s:=0;
  for j:=1 to k do
    if (j<>) and (s<a[i,j]) then s:=a[i,j];
  s:=s+a[i,j];
  if (i=1) or (s<t) then
    begin
      il:=i;
      t:=s;
    end;
  end;
writeln (il);
end.

```

212 – масала

Алгоритм «Тартиб индекслари»

Ҳаммасидан осони «изланаётган тартиб индекслари» учун қўшимча $I[1:N]$ массив ҳосил қилиш, аввал уни $1,2,\dots,N$ сонлари билан тўлдириш, кейин берилган A массивни, қандайдир усул билан тартибга келтириш, худди шундай тартибда I массив элементларининг ўрнини алмаштириш керак. A тартибга келтирилган вақтда, I изланаётган массив бўлади.

Келтирилган дастурда худди шу иш бажарилган, бу ерда тартибга келтиришнинг энг оддий усули ишлатилган: A массивда энг кичик элемент бор ва бу элемент биринчиси билан ўрнини алмаштириб боради. Кейин бу жараён $A[2:N]$ кесмага ва ҳоказо қўлланилади.

Агар дастурни мураккаблаштириш керак бўлса, I массивсиз ишни бажариш мумкин, лекин A массив элементлари ўзгармаслиги керак. Бунда навбатдаги минимал элемент (MI) индексини чоп этиш ва эслаб қолиш етарли. Бунинг учун, албатта, биринчи шундай элементнинг индексини ва биринчи максимал элемент индексини олдиндан топиб қўйиш керак.

Дастур

```

Program Tartib_indekslari;
const NN=100;
label 1,2;
var

```

```

j,i,m,mi,ma,k,n :integer;
A : array [1..NN] of real;
begin
  Writeln('N:=');
  readln(n);
  for i:=1 to n do
    begin
      writeln ('a[',i,']:=');
      readln (a[i]);
    end;
    mi:=1; ma:=1;
    for k:=1 to n do
      begin
        if a[k]<a[mi] then mi:=k;
        if a[k]<a[ma] then ma:=k;
      end;
    writeln (mi);
    for m:=2 to n do
      begin
        i:=ma;
        for k:=1 to n do
          begin
            if (a[k]<a[mi]) or (a[k]=a[mi]) and (k<=mi)
              then goto 1;
            if a[k]=a[mi] then
              begin
                i:=k;
                goto 2;
              end;
            if a[k]<a[i] then
              i:=k;
          1 : end;
          2 : mi:=i;
          writeln (mi);
        end;
      end;
    end.

```

213 – масала

Алгоритм «Ноллаштириш»

Масалани ечиш учун энг осон йўл, $V[1:m]$ ва $C[1:n]$ иккита ёрдамчи массивларни киритиш. Кейинчалик A массив элементлари кўздан кечирилади бориб, V ва C массивларда ноллар учраган сатр ва устунлар белгилаб борилади, яъни: **if** $a[i,j] = 0$ **then** $b[i] = 1 : c[j] = 1$ бажарилади.

Энди, A массив иккинчи марта кўздан кечириляётганда, V ёки C массивларда белгиларга эга бўлган, сатр ва устунларда жойлашган элементлар ноллар билан алмаштирилади, яъни

if $b[i] = 1$ **or** $c[j] = 1$ **then** $a[i,j] = 0$

Фақат битта $C[1:n]$ ёрдамчи массивдан фойдаланадиган бошқа ечимни таклиф қилиш мумкин. Бу ҳолда A массив сатрлар бўйича кўздан кечирилади. Нол элементлари бўлган устунлар, аввалгидай C массивда белгиланади, кўриляётган сатрда нол учрагани эса Z ўзгарувчида қаралади, яъни:

if $a[i,j] = 0$ **then** $c[j] = 1 : Z=1$

i сатр қараб чиқилгач, агар $z=1$ бўлса, сатр ноллаштирилади. Иш сўнгида C массив қаралади ва агар $c[j] = 1$ бўлса, j устунлар ноллаштирилади.

Бу масалани ечишга хос хатоликлар :

1) A массив қаралади ва агар $a[i,j]$ элемент нолга тенг бўлса, i сатр ёки j устун ёки униси ҳам, буниси ҳам ноллаштирилади. Кейинги қарашларда ноллаштирилган элементлар бошланғич нолга тенг элемент, деб қабул қилинади ва ортиқча ноллаштиришларни ҳосил қилади.

2) A массив қаралади ва агар $a[i,j]=0$ бўлса, навбатдаги $c[k]$ элементга j киритилади. Бундан битта j рақам кўп марта киритилиши ва C массив, шартга зид равишда, $m*n$ ўлчамда талаб қилиниши мумкин.

Дастур

```
Program Nollashtirish;
```

```
const NN=20;
```

```
MM:=20;
```

```
var
```

```
i,j,m,n :integer;
```

```
Z      : boolean;
```

```

A : array [1..MM,1..NN] of real;
C : array [1..NN] of boolean;
begin
  writeln ('M,N:=');
  readln (m,n);
  for i:=1 to m do
    for j:=1 to m do
      begin
        write ('a[',i,',',j,']:=');
        readln (a[i,j]);
      end;
    for j:=1 to n do C[j]:=false;
    for i:=1 to m do
      begin
        z:=false;
        for j:=1 to n do
          if a[i,j]=0 then
            begin
              z:=true;
              c[j]:=true;
            end;
          if z then for j:=1 to m do a[i,j]:=0;
        end;
        for j:=1 to n do
          if c[j] then for i:=1 to m do a[i,j]:=0;
        for i:=1 to m do
          begin
            for j:=1 to n do writeln(a[i,j]:4);
            writeln;
          end;
        end.

```

214 – масала

Алгоритм «Улгуржи харид»

Улгуржи харид орттирмаларсиз қуйидагича топилади:

$n1 = n \div 144; m = n - n1 * 144$

$n2 = m \div 12; n3 = m - n2 * 12$

уни фақат икки йўл билан арзонлаштириш мумкин ортиқча боғлам олиб, ортиқча жуфтлик олмаслик ёки ортиқча қути олиб, боғлам ҳам, жуфтлик ҳам олмаслик керак. Агар $n3 * 1.05 > 10.25$ бўлса ортиқча:

$n2 = n2 + 1; n3 = 0$

боғлам олиш керак.

Агар ҳосил бўлган харидда (эски ёки ўзгартирилганида) $n2 * 10.25 + n3 * 1.05 > 114.00$ бўлса, $n1 = n1 + 1; n2 = 0; n3 = 0$ ни ҳосил қилиш керак.

Бу ечим ҳар хил нархларда (кути 12 боғламдан, боғлам 12 жуфтдан арзон бўлган ҳолларда) яроқлидир.

Тест мисоллар

n	n1	n2	n3
9	0	0	9
10	0	1	0
131	0	11	0
134	1	0	0

Дастур

```
Program Ulgurji_savdo;
```

```
var
```

```
    n1,n2,n3,m,n :integer;
```

```
begin
```

```
    writeln ('N:=');
```

```
    readln (n);
```

```
    n1:=n div 144;
```

```
    m:=n-n1*144;
```

```
    n2:=m div 12;
```

```
    n3:=m-n2*12;
```

```
    if (n3*1.05 )>10.25 then
```

```
        begin
```

```
            n2:=n2+1;
```

```
            n3:=0
```

```

    end;
    if (n2*10.25 + n3*1.05 )>114 then
    begin
        n1:=n1+1;
        n2:=0;
        n3:=0
    end;
    writeln (n1,' ',n2,' ',n3 );
end.

```

215 – масала

Алгоритм «Тўнтарилган сонлар»

Бу масала (N^2 амалда бажарилиш учун) маълум бир фикрлашни талаб этади. Тадқиқ қилинадиган кесмани унинг маркази билан бериш (бу N та вариантни беради) ва марказдан иккала томонга симметрик элементларни таққослаш керак (бу N та гуруҳдан катта бўлмаган амалларни талаб қилади). Шундай қилиб, ечим N^2 амалда (доимий кўпайтувчи аниқлигига) топилади. Ечимни расмийлаштиришда кесма маркази ўрнига (у элементга ҳам, элементлар ўртасига ҳам тўғри келиши мумкин) марказ билан қўшни бўлган икки элемент рақамини $LN < PN$ олиш кулай. Бу рақамлар тушунарли тарзда силжийди:

$$(LN, PN) = (1, 2), (1, 3), (2, 3), (2, 4), \dots$$

таққосланадиган элемент рақамлари бошда $L=LN : P=PN$, деб олинади ва кейин таққосланадиган $A[L]$ ва $A[P]$ элементлар тенг бўлгунча ёки L ва P нуқталардан биттаси $(1, N)$ чегарадан чиққунча, ҳар қадамда $L=L-1 : P=P+1$ бажарилади. Шунда (LN, LP) маркази максимал кесма $M=P-L-1$ узунликда бўлади.

Дастур

```

Program To'ntarilgan_sonlar;
const NN=100;
var
    l,i,ln,pn,p,m,max,n :integer;
    z :boolean;
    A : array [1..NN] of real;

```

```

begin
  writeln ('n:=');
  readln(n);
  for i:=1 to n do
    begin
      writeln ('a[',i,']:=');
      readln (a[i]);
    end;
  max:=1; z:=true; ln:=1; pn:=2;
  while 2*(n*pn+1)+1>max do
    begin
      l:=ln; p:=pn;
      while (l>=1) and (p<=n) and (A[l]=A[p]) do
        begin
          l:=l-1; p:=p+1;
        end;
      m:=p-l-1;
      if (max<m) then max:=m;
      if z then pn:=pn+1 else ln:=ln+1;
      z:=NOT z ;
    end;
  writeln (max);
end.

```

216 – масала

Алгоритм «Икки марта монотон»

$i=1, j=n$ деб олинади, $a[i,j]$ элемент (аввал бу, кейин навбатдагиси) x сони билан таққосланади.

агар $a[i,j] = x$ бўлса, жавоб топилади,

агар $a[i,j] < x$ бўлса, $i=i+1$ деб олинади,

агар $a[i,j] > x$ бўлса, $j=j-1$ деб олинсин.

Охирги икки ҳолда $i \leq m$ ва $j \geq 1$ қолганлигини текшириш керак. Агар қолган бўлса, таққослашга қайтиш, акс ҳолда «йўқ» деб чоп этиш керак. Ҳар бир қадамда i оширилиб, ёки j камайтирилиб борилади. Демак, қадамлар $m+n$ дан кўп бўлмайди.

Бу алгоритмнинг тўғрилигига ишонч ҳосил қилиш қийин эмас. Ҳақиқатдан ҳам, агар $a[i,j] < x$ бўлса, x сони $a[i:m,1:j]$ массивнинг i сатрида йўқ ва бу сатрни ташлаш мумкин. Агар $a[i,j] > x$ бўлса, худди шу сабабга кўра j устунни ташлаш мумкин.

Дастур

```
Program Ikki_marta_monoton;  
const MM=20;  
        NN=20;  
label 1;  
var  
        i, j, m, n, x :integer;  
        A : array [1..MM,1..NN] of real;  
begin  
        writeln ('m,n:=');  
        readln (m,n);  
        writeln ('x:=');  
        readln (x);  
        for i:=1 to m do  
        for j:=1 to n do  
                begin  
                        writeln ('a[' ,i ,',',j ,']:=');  
                        readln (a[i,j]);  
                end;  
        i:=1; j:=n;  
        while (i<=m) and (j>=1) do  
                begin  
                        if a[i,j]=x then goto 1;  
                        if a[i,j]<x then i:=i+1 else j:=j-1;  
                end;  
        writeln ('Йўқ');  
1: writeln (i, ', ',j);  
end.
```

217 – масала

Алгоритм «Тетраэдрлар»

Ечим икки қисмга бўлинади: М1 билан мос келтириш учун N тетраэдрнинг N' қиррасини ва М1 да қолдирадиган N тетраэдрнинг бурилишларини танлаш. Биринчиси 4 усул билан, иккинчиси 3 та усул билан бажарилади. Фақат, N' танланганидан кейин N нинг қолган қирраларини қайси биттасидандир бошлаб маълум тартибда (масалан, соат йўналишига тескари) ёзиб олиш, ечимнинг иккинчи қисмини функция кўринишида расмийлаштириш керак (дастурда келтирилган).

Ечимдан ҳамма 12 вариант бевосита ёзиб олинадиган жавоб қониқарли эмас.

Дастур

Program Tetraedrlar;

var

m1,m2,m3,m4,n1,n2,n3,n4 :integer;

function DA (i,j,k,l:integer) : boolean;

begin

DA:=((m1=i) **and** (((m2=j) **and** (m3=k) **and** (m4=i))
or ((m2=l) **and** (m3=j) **and** (m4=k))
or ((m2=k) **and** (m3=l) **and** (m4=j))))

end;

begin

writeln ('m1,m2,m3,m4:=');

readln (m1,m2,m3,m4);

writeln ('n1,n2,n3,n4:=');

readln (n1,n2,n3,n4);

writeln;

if DA (n1,n2,n3,n4) **or** DA(n2,n1,n4,n3) **or** DA(n3,n1,n2,n4) **or**
DA(n4,n1,n3,n2) **then** writeln('Ҳа')
else writeln('Йўқ');

end.

218 – масала

Алгоритм «Каср даври»

Ечим шартда кўрсатилмаган кўшимча талабларга қаттиқ боғлиқ. «М» натурал сонни «N» натурал сонга бўлганда бўлинманинг бутун қисми ва қолдиқ ҳосил бўлади:

$$i=M/N : k=M-i*N$$

Масалани ечиш учун аввал M/N касрдан M суратни қолдиққа алмаштириб, бутун қисмини йўқотамиз:

$$M = M - M/N * N$$

Энди, ҳосил бўлган касрнинг i бўлинма ва M қолдиқ рақамларини кетма-кет ҳосил қилиш мумкин:

$$i = 10 * M/N : M = 10 * M - i * N \text{ ва ҳоказо.}$$

Ҳар бир қолдиқ N дан ошмайди. Демак, ҳар хил қолдиқлар N дан катта бўлмайди ва улар такрорлана бошлайди. Қолдиқ қачон такрорланса, бўлинмалар ҳам такрорлана бошлайди, давр бошланади. Такрорлашни илғаб олиш учун, энг осони $D[1:N]$ массивни ташкил қилиш ва унга, ҳосил бўлиш тартибига кўра, қолдиқларни ёзиш ва ҳар бир янги қолдиқни барча олдингилари билан таққослаб бориш керак.

Баён этилган ечим тўғри, лекин яхши эмас. Ҳисоб бўйича K -қолдиққа K таққослаш, ҳамма қолдиқларга эса $N^2/2$ таққослаш кетади. Замонавий ЭҲМ ларда 10^8 тартибли бутун M ва N ни ҳосил қилиш осон, лекин 10^{16} амалларни бажариш учун йиллар кетади, шу сабаб N сони учун D массивни ташкил қилишга келишилган экан, уни тозалаймиз ва навбатдаги M қолдиқнинг пайдо бўлишини $D[M]$ элементда белгилаймиз. У пайтда K қолдиқ «янгилиги»ни текшириш битта таққослашни эгаллайди.

Лекин бу ечим ҳам жиддий камчиликка эга. N сони шундай катта бўлиши мумкинки, бунда N та амални бажариш мумкин бўладию, лекин N та элементдаги массивни тезкор хотирага киритиш мумкин бўлмай қолади. Қолдиқлар учун массивни ташкил қилмай, даврни топишга уриниб кўрамиз. Дастлаб аввалги каби «М» дан «N» га бўлинадиган қисмини ажратамиз. Кейин бўлинманинг N рақамини ўтказамиз. Энди, давр бошлангани маълум бўлгач, битта ягона қолдиқни эслаб қоламиз ва бўлинманинг рақамлари такрорлана бошланмагунича чоп этамиз. Бу ишлар таклиф этилган дастурда келтирилган. Бу дастур олдингисидан қисқа.

Дастур

```
Program Kasr_davri;  
var  
    m,n,i,j,k :integer;  
begin  
    writeln ('m,n:=');  
    readln (m,n);  
    writeln;  
    M:=m-(m div n)*n;  
    k:=1;  
    while (k<=n) or (j<>m) do  
    begin  
        if k=n then j:=m;  
        i:= 10 * m div n;  
        m:=10 * m-i * n;  
        if k>=n then write(i);  
        k:=k+1;  
    end;  
    writeln ;  
end.
```

219 – масала

Алгоритм «Arra»

Масала тушунтириш талаб қилмайди.

Дастур

```
Program Arra;  
const mm=100;  
var  
    i,k,j,m :integer;  
    x : array [1..mm] of real;  
begin  
    writeln ('m:=');  
    readln (m);  
    for i:=1 to m do readln(x[i]);
```

```

j:=1; k:=1; i:=1;
while i+1<m do
    begin
        if (x[i]<x[i+1]) and (x[i+1]>x[i+2]) then
            begin
                i:=i+2; j:=j+2;
                if k<j then k:=j
            end
        else
            begin
                i:=i+1; j:=1;
                if i+k>=m then i:=m-1;
            end;
    end;
writeln;
writeln (k);
end.

```

220 – масала

Алгоритм «Ноллар охирига»

Масала икки циклни киритиш билан осон ечилади: биринчиси нолга тенг бўлмаган элементларни массив бошига ёзади, иккинчиси эса қолдиқни ноллар билан тўлдиради. Лекин бу цикллارни биргаликда ҳам ташкил этиш мумкин.

Дастур

```

Program Nollar_oxiriga;
const nn=100;
var
    i,n,j : integer;
    x : array [1..nn] of integer;
begin
    write ('n=', n);
    readln (n);
    for i:=1 to n do readln(x[i]);

```

```

j:=0;
for i:=1 to n do
if x[i] <>0 then
begin
j:=j+1;
if i<>j then
begin
x[j]:=x[i]; x[i]:=0;
end;
end;
for i:=1 to n do writeln(x[i], '');
writeln;

```

end.

221 – масала

Алгоритм «Ҳар хил рақамли сонлар»

Дастур керакли даражада тушунарли.

Дастур

```

Program Har_xil_raqamli_sonlar;

```

```

var

```

```

i,m,j,k:integer;

```

```

begin

```

```

for i:=1 to 9 do

```

```

for j:=0 to 9 do if i<>j then

```

```

for k:=0 to 9 do if (k<>i) and (k<>j) then

```

```

for m:=0 to 9 do

```

```

if (m<>i) and (m<>j) and (m<>k) then

```

```

writeln (((i10+j)*10+k)*10+m);

```

```

end.

```

222 – масала

Алгоритм «0,1,2 ларнинг ўрнини алмаштириш»

Бу масалани ечишда ҳеч ниманинг ўрнини алмаштириш керак эмас. Массивда нечта ноллар, бирлар ва иккилар борлигини санаш ва массивни талаб бўйича тўлдириш керак.

Дастур

```
Program O'rin_almashtirish;
const nn=100;
type el=0..2;
var
    i,n :integer;
    x : array [1..nn] of el;
    a : array [el] of 0..nn;
begin
    readln (n);
    writeln ('n:=',n);
    for i:=1 to n do readln (x[i]);
    writeln;
    a [0]:=0; a[1]:=0;a[2]:=0;
    for i:=1 to n do a[x[i]]:=a[x[i]]+1;
    for i:=1 to n do
        if i<=a[0] then x[i]:=0
            else if i<=n-a[2] then x[i]:=1 else x[i]:=2;
    for i:=1 to n do writeln (x[i]);
end.
```

223 – масала

Алгоритм «Арифметик амаллар».

Бу масала ҳам барча ўрин алмаштиришларни, фақат такрорлаш билан ҳосил қилишга бағишланган. Берилган ифодани қуйидаги кўринишда ёзамиз :

$$w = (((((1 \ a_2 \ 2) \ a_3 \ 3) \ a_4 \ 4) \ a_5 \ 5) \ a_6 \ 6$$

бу ерда A[6] массив элементлари арифметик амал белгиларини билдиради. +,-,*,÷ белгилар 1,2,3,4 қийматлар билан ифодаланеди, деб келишиб оламиз. Вариантлар кўп бўлиши мумкинлиги учун ($4^5 > 1000$), ҳисоблашларни тежаб ташкил қилишга ҳаракат қиламиз. Бунинг учун, w

ни ҳисоблаб бориб, оралиқ натижаларни $B[6]$ массив элементлари ($B_1:=1$:
 $B_2:=B_1$ a_2 2 : ... : $B_6:=B_5$ a_6 6 ва $w:=B_6$) деб олиб хотирлаймиз. Кейин
вариантларни кўриб чиқа бориб, аввал $a_6=1,2,3,4$ ни алмаштирамиз. Ҳар
бир бундай ўзгаришда фақат B_6 ни ҳисоблаб чиқиш керак бўлади. Кейин
 $a_5=a_5+1$ ни оширамиз, B_5 ни қайта ҳисоблаб, $a_6=1,2,3,4$ ва ҳоказо.

Дастур

```

Program Arifmetik_amallar;
const m=1; n=9;
label 5;
var
    j, k, y :integer;
    A,B : array [1..n] of integer;
begin
    b[i]=1; k:=0; a[2]:=0;
    for i:=3 to n do a[i]:=4
5: for i:=n downto 2 do
    if (a[i]=4) then a[i]:=1
        else
            begin
                A[i]:=A[i]+1;
                Y:=B[i-1];
                case (A[i]) of
                    1:B[i]:=y+i;
                    2:B[i]:=-y-i;
                    3:B[i]:=y*i;
                    4:B[i]:=y div i;
            end;
    for j:=j+1 to n do b[j]:=b[j-1]+j;
    if (b[n]<>M) then goto 5; k:=k+1;
    for j:=2 to n-1 do write (('')); write ('1');
    for j:=2 to n do
        begin
            if (j>2) then write ('')
            case a[j] of
                1:write ('+');
                2: write ('-');
                3: write ('*');
                4: write ('%');
        end

```

```

                end;
                writeln (j);
            end;
        writeln ('=',m); goto 5;
    end;
    writeln(k);
end.

```

224 – масала

Алгоритм «Тез даража»

Одатдаги a^k ни ҳисоблашда дастлаб қиймати 1 га тенг b ўзгарувчи киритилади ва k ўп марта

$$K:=K-1; b:=b*a$$

операторлар бажарилади.

K ўзгарувчи нолга тенг бўлган пайтда (бу k цикли талаб қилади), b изланаётган a^k миқдорга тенг бўлади. Ҳисоблашни қисқартиришнинг моҳияти қуйидагича. Дастлаб, $b:=1$. Агар K тоқ бўлса, аввалгидай

$$K:=K-1; b:=b*a$$

операторлар бажарилади.

Агар k жуфт бўлса, унда,

$$a^k = (a^2)^{k/2}$$

айниятдан фойдаланиб, $k:=k/2$; $a:=a*a$ алмаштириш бажарилади.

Ниҳоят, « K » ўзгарувчи нолга тенг бўлган пайтда, « b » ўзгарувчи изланган миқдорга тенг бўлади. Исбот учун изланаётган a^k миқдорни w деб белгилаймиз, унда

$$b:=1; a^k \cdot b=w \quad (*)$$

агар K тоқ бўлса,

$$K:=K-1; b:=b \cdot a$$

шакл алмаштиришни бажариб, (*) тенгликни бузмаймиз. Агар k жуфт бўлса:

$$K:=K/2; a:=a \cdot a$$

алмаштириш ҳам уни бузмайди. K нолга тенг бўлган пайтда, (*) тенглик

$$a^0 \cdot b=w, \text{ яъни } b=w,$$

тенгликка айланади ва, демак, b изланаётган миқдорга тенг бўлади.

Дастур

```
Program Tez_daraja;  
var  
    a, b : real;  
    k, n : integer;  
begin  
    readln (a,k);  
    write(a, '^', k, '=');  
    b:=1;  
    while k>0 do  
        begin  
            n:=k div 2; if (n+n<k) then b:=b*a;  
            k:=n; a:=a*a;  
        end;  
    writeln (b);  
end.
```

225 – масала

Алгоритм «Ҳар хил сонлар»

Дастур осон ўқилади.

Текширилаётган элемент биринчи такрорлашгача олдингиси билан эмас, балки кейингиси билан таққосланса, дастур ўқилиши яна ҳам тезлашади. Бундан такрорланишлар сони $m*m$ дан $m*k$ гача тушади. Бу ерда, массивда m – ҳамма сонлар, k – эса ҳар хил сонлар миқдорини билдиради..

Дастур

```
Program Har_xil_sonlar;  
const n=100;  
var  
    i,j,m,S : integer;  
    b : boolean;  
    A : array [1..n] of integer;  
begin  
    readln (m);
```

```

writeln ('m:= ',m);
for i:=1 to n do
    begin
        readln (a[i]);
        writeln ('a[',i, ']:= ',a[i]);
    end;
S:=0;
for i:=1 to m do
    begin
        b:=false;
        j:=i+1;
        while (j<=m) and not b do
            begin
                b:=b or (a[i]=a[j]);
                j:=j+1;
            end;
        if not b then S:=S+1;
    end;
writeln ('S=',S);
end.

```

226 – масала

Алгоритм «Минимумларнинг максимуми»

Келтирилган дастур, биринчи сатр ва сатрнинг ҳар бир элементи алоҳида кўздан кечиришни талаб қилинмайдиган қилиб тузилган. Бундан ташқари, сатрни ишлаб чиқиш навбатдаги сатр изланаётган элементга эга эмаслиги маълум бўлиши билан тўхтатилади.

Дастур

```

Program Minimumlar_maksimumi;
const mm=10; nn=20;
label 1;
var
    i,j,n,m,k :integer;
    min,max :integer;

```

```

    x : array [1..mm,1..nn] of integer;
begin
    readln (m,n);
    writeln ('m:= ',m, 'n:= ',n);
    for i:=1 to m do
    for j:=1 to n do
    begin
        writeln ('x['i','j,']:=');
        readln (x[i,j]);
    end;
    for i:=1 to m do
        begin
            for j:=1 to n do
                begin
                    if (i>1) and (x[i,j])<=max) then goto 1;
                    if (j=1) or (x[i,j])<=min) then min:= x[i,j];
                end;
                max:=min; k:=i;
            1 : end;
            writeln ('k',k);
        end.

```

227 – масала

Алгоритм «Спирал».

Спирал тармоғи тўртта тўғри чизикли қисмга бўлинади: чапдан ўнгга горизонтал, юқоридан пастрга вертикал, ўнгдан чапга горизонтал ва қуйидан юқорига вертикал. Дастурда уларнинг ҳар бири алоҳида тўлдирилади ва бир қисмнинг иккинчисига «сирпалиб ўтиши» ишлатилади. Унинг тугаши охири $k=n*n$ сонинг киритилиши билан текширилади.

Дастур

Program Spiral;

const nn=19;

var

```

i,n,j,k   : integer;
function MOV :boolean;
A : array [1..nn,1..nn] of integer;
begin
MOV:=false;
if k<=n*n then
    begin
        a[i,j]:=k; k:=k+1;
        MOV:=true
    end;
end;
begin
writeln ( 'n:=' ,n);
readln (n);
k:=1; i:=1; j:=1;
repeat
    while MOV and (i+j<n+1) do j:=j+1; k:=k-1;
    while MOV and (i<j) do i:=i+1; k:=k-1;
    while MOV and (i+j<n+1) do j:=j-1; k:=k-1;
    while MOV and (i>j+1) do i:=i-1; k:=k-1;
until k:=n*n;
for i:=1 to n do
    begin
        for j:=1 to n do
            writeln ( a[i,j]:4);
            writeln ;
    end;
end.

```

228 – масала

Алгоритм «Ички тўплам бўйича йиғинди».

Бу масала ҳамма ички тўпламларни санаб ўтишга доир. Ташқаридан қараганда у вариантларни кўриб чиқиш масаласига ўхшайди, лекин унга қараганда анча содда, b -натурал сон ва b_i -унинг иккилик хоналари ($b_i=0$ ёки $b_i=1$) бўлсин:

$$b = b_1 + 2b_2 + \dots + 2^{n-1} * b_n.$$

$b = 1, 2, \dots, 2^{n-1}$ қийматлар кетма-кетлиги бўйлаб «b» ўтганда, бирга тенг b_i элементларнинг i индекслар тўплами $\{1, 2, \dots, n\}$ тўпланинг ҳамма (бўш бўлмаган) ички тўпланиларидан ўтади. Шунинг учун дастурда P массив киритилган. Унинг элементлари билан «b» сонининг иккилик разрядлари каби иш олиб борилади.

Дастур

```
Program Ichki_to'plam_summasi;  
const nn=100;  
label 1;  
var  
    m,n,i,s :integer;  
    a : array [1..nn] of integer;  
    b : array [1..nn] of boolean;  
begin  
    writeln ('n,m=');  
    readln (n,m);  
    for i:=1 to n do  
        begin  
            writeln ('a['i,]= ');  
            readln (a[i]);  
            b[i]:=false;  
        end;  
    s:=0;  
    repeat  
        for i:=1 to n do  
            if (b[i]) then  
                begin  
                    b[i]:=false; s:=s-a[i]  
                end  
            else  
                goto 1;  
            1: b[i]:=true; s:=s+a[i];  
    until s:=m;
```

```
for i:=1 to n do
    if (b[i]) then write(I:4);
writeLn;
end.
```

229 – масала

Алгоритм «Лабиринт»

Яхши маълум бўлган бу масала, бу ерда бироз ўзгартирилган. Ечим иккита қисмга бўлинади: чиқиш йўлини излаш ва «тескари йўл»ни (йўловчининг чиқишдан бошлаб, бошланғич ҳолатигача) чоп этишдан иборат.

Биринчи қисмнинг оддий ечими қуйидагича баён этилади: Йўловчининг бошланғич нуқтаси бўлган $a[i,j]$ катакка 2 сони ёзилади ва $k=2$, деб олинади. А лабиринтнинг ҳамма катаклари кўздан кечириб чиқилади. Ёзилган ҳар бир катак учун унинг тўртта қўшниси кўздан кечирилади. Агар унга қўшни бўлган тўртта катакдан, ҳеч бўлмаганда, биттасида k (ҳозир 2 га тенг) ёзилган бўлса, кўрилатган a катакка $k+1$ сони ёзилади. Энди $k=k+1$ оширилади ва A нинг ҳамма катаклари қайтадан кўздан кечирилади. Агар сон чегара катакка ёзилган бўлса, (йўл топилган) ёки A нинг ҳамма катакларини кўздан кечиришда, у биронта ҳам каттакка ёзилмаган бўлса (йўл йўқ), жараён тугайди. Қисқа йўлда нечта катак бўлса, массивнинг ҳаммасини қараб чиқиш ҳам шунча бўлади.

Олдинги алгоритмни яхшилаш осон. Аввалгидек, ҳар қадамда A лабиринтнинг ҳамма катаклари кўздан кечирилади. Агар катакда, нол ёзилган бўлсаю, қандайдир унга қўшни катакда $K \geq 2$ сон бўлса, a катакка $k+1$ сони ёзилади. Бу масала биринчи қисмининг ечими эканлиги аниқ. Лекин, агар омад чопса, бу ерда ечим тез топилади.

Бошланғич катакдан бошлаб (бошда 2 ёзилган катак), биринчи бўш (яъни, Олик катак) қўшнини изловчи дастур самарали бўлади. Қўшни бўш катакка 3 сони ёзилади ва 4 сонини ёзиш учун унинг бўш қўшниси изланади ва ҳоказо. Чегарага чиқилганда (йўл топилди) ёки бўш қўшнилар бўлмаганда берк катак жараён тўхтатилади. Агар берк катак бошланғич катакда пайдо бўлса, чиқиш йўқ. Агар берк катак бошқа бўлиб, унга $k>2$ сони ёзилган бўлса, унга бир сонини ёзиш (уни ёпиш) ва қўшни, $k-1$ сонли катакка ўтиш керак. Бундай катак бор ва у ягона. Бу ечимда вариантларни

кўриб чиқишнинг умумий схемасини топиш осон. Лабиринт тўғрисидаги масала ечимини рекурсив ёзиб, сезиларли қисқартириш мумкин. Бу иш таклиф этилган дастурда бажарилган. Лекин бундай дастурни ўқишдан кўра ёзиш осон.

Келтирилган алгоритмлар (биринчисидан бошқа) энг қисқа йўлни бермаслиги ҳам мумкин. Лабиринтда қисқа йўлни тежамли излаш учун кўздан кечирилиши керак бўлган катак (x,y) координаталари рўйхати учун махсус X ва Y массивларни ташкил этиш мумкин. Бундай усул кенглиги (эни) бўйича излаш дейилади.

Аввал X,Y га йўловчининг бошланғич катак координаталари киритилади. Ҳар қадамда X ва Y массивлардан, кўздан кечириш учун навбатдаги катак («b» рақамли) координаталари олинади, унинг бўш кўшнилари эса ихтиёрий тартибда X,Y рўйхат давомида «e» рақам билан ёзиб қўйилади. Шундай қилиб, рўйхат бошидан ишлаб чиқилади ва охиридан узайтирилади.

Излаш лабиринт чегарасида бўш катакка ётиш билан ёки X, Y рўйхатни тугатиш билан (чиқиш бўлмаса) тамомланади.

Тескари йўлни излаш ва чоп этиш рекурсивмас дастурларда бир хил, рекурсивларда эса автоматик бажарилади.

Дастур

Program Labirint;

const mm=15; nn=15;

var

 i,n,j,m :integer;

 chik : boolean;

 a : **array** [1..mm,1..nn] **of** byte;

procedure L (i,j:integer);

begin

if not chik **then**

if a[i,j]=0 **then**

begin

if i=1 **or** i=m **or** j=1 **or** j=n **then** chik:=true;

 a[i,j]=1; L(i,j-1); L(i,j+1); L(i-1,j); L(i+1,j);

if chik **then** writeln (i, ' ', j);

end;

```

end;
begin
    writeln ('m,n=');
    readln (m,n);
    for i:=1 to m do
        for j:=1 to n do
            begin
                writeln ('a[' ,i ,',',j ,']=');
                readln (a[i,j]);
            end;
        writeln;
        writeln ('i,j=');
        readln (i,j)
        writeln;
        chik:=false;
        l(i,j);
        if not chik then writeln ('chiqish yoq');
    end.

```

230 – масала

Алгоритм «Мукаммал сонлар».

Натурал i сон мукаммал бўлишини ҳал қилиш учун ҳамма $j = 1, 2, \dots, i-1$ сонларни кўриб чиқиш, уларнинг қайси бири i сонининг бўлувчиси бўлишини аниқлаш ва бу бўлувчиларни қўшиш керак.

Бўлувчига «номзодлар»ни $i-1$ гача эмас, балки \sqrt{i} гача кўриб чиқиб, дастур бажарилишини тезлаштириш мумкин. Бунинг учун $k = i / j$ ($k \geq j$) ни топиш ва агар i сонининг бўлувчиси j бўлса, фақат j ни эмас,

балки k ни ҳам ҳисобга олиш керак. Фақат биронта ҳам бўлувчи икки марта олиншига ($j=k$ да) йўл қўймаслик керак. Бу алгоритм таклиф қилинган дастурда келтирилган.

Дастур

```

Program Mukammal_sonlar;

```

```

var

```

```

    m,i,j,k,s :integer;

```

```

begin
  write ('m:=');
  readln (m);
  writeln ('program get m=',m);
  for i:=2 to m do
    begin
      S:=1; j:=1;
      repeat
        J:=J+1;
        K:=i*div j;
        if (i=k*j) and (j<=k) then
          begin
            s:=s+j if j>k then s:=s+k;
          end;
        until j>=k
        if s=i then writeln (i);
      end;
    end;
  end.

```

231 – масала

Алгоритм «Ҳосил қилинмайдиган сон».

Бу масала биринчи қарашдагидан кўра қизиқроқ, чунки у $n*n$ (хатто $n*\log n$) амалда ечилади. Мумкин бўлган ечим қуйидагича $S=1$ деб олинади, P массивда қандайдир $P(j)\leq S$ элемент борлиги аниқланади. Агар бундай элемент бўлмаса, « S » ечим бўлади. Агар бундай элемент топилса, унинг қиймати S га қўшилади, яъни $S=S+P[j]$ Массивдан $P(j)$ олиб ташланади ва яна қолган элементлар ичидан « S » нинг янги қийматидан катта элемент борлиги изланади ва ҳоказо. Индукция усули бўйича худди шундай тарзда биринчи ташкил этилмайдиган сонни топишни исботлаш мумкин. Ҳақиқатда навбатдаги $i=1,2,\dots$, қадамда P массив элементларини $P(i)$ дан бошлаб кўриб чиқиш, топилган $P(j)$ элементни эса $P(i)$ элементга алмаштириш керак.

Ҳаракатлар (амаллар) сонини тежаш мақсадида (массивда сонлар энг ёмон жойлашган ҳолда) олдиндан P массивнинг фақат навбатдаги

элементини таққослаш керак бўлади. Баён этилган алгоритм таклиф қилинган дастурда амалга оширилган.

Дастур

```
Program Hosil_qilinmaydigan_son;  
const NN=10;  
label 1,2;  
var  
    n,i,j,s    : integer;  
    P : array [1..NN] of integer;  
begin  
    writeln ('n:=');  
    readln (n);  
    for i:=1 to n do readln (P[i]);  
    writeln;  
    S:=1;  
    for i:=1 to n do  
        begin  
            for j:=i to n do if P[j]<=s then goto P1;  
            goto 2;  
1: s:=s+P[j]; P[j]:=P[i];  
        end;  
    2 : writeln(s);  
end.
```

232 – масала

Алгоритм «Дўмбира».

Қуйидаги оддий ечим таклиф этилади. К - «кичикликка номзод» вектор рақами, р-навбатдаги вектор рақами бўлсин. Бошида к =1 ва р=2, деб олинади.

Агар қандайдир бир лаҳзада қуйидагилар аниқланса:

- 1) $X_k \leq X_r$ бўлса, р бирга оширилади;
- 2) $X_k > X_r$ бўлса, к р га алмаштирилади ва р бирга оширилади.

Агар шундан кейин $p \leq 12$ бўлса, X_k ва X_r векторлар янги жуфтани таққослашга ўтиш керак. Агар $p > 12$ бўлса, сонлар ечим бўлади. Бу алгоритм таклиф этилган дастурда амалга оширилган.

Бошқа хил ечим ҳам бор. У баъзи бир ҳолларда тезроқ ҳосил қилиниши мумкин. Унга кўра «кичикликка номзод» векторлар рўйхати учун қўшимча $L[1:12]$ массив ташкил қилинади. (Аниқроғи-«номзод»лар бошланадиган рақамлар рўйхати учун). Бошда бу 1 дан 12 гача рақамлар бўлади. Шундан сўнг уларнинг ичидан биринчи элементи кичик бўлганлари қолдирилади, улардан иккинчи элементи кичик бўлганлари ва хоказо биронта ҳам номзод қолмагунча ёки минимал векторларга номзод ҳамма 12 та элемент қараб чиқилмагунча давом эттирилади.

Дастур

```
Program Do'mbira;  
const N=12;  
label 1;  
var  
    i,j,p,l,k :integer;  
    A : array [1..n] of real;  
begin  
    for i:=1 to n do  
        begin  
            writeln ('A[',i,']:=');  
            readln (A[i]);  
        end;  
    k:=1;  
    for P:=2 to n do  
        begin  
            for i:=0 to n-1 do  
                begin  
                    j:=k+i; if j>n then j:=j-n;  
                    l:=P+j; if l>n then l:=l-n;  
                    if A[l]<A[j] then k:=p;  
                    if A[l]<>A[j] then goto 1;  
                end;  
        end;  
end;
```

```

1: end;
writeln;
writeln (k);

```

end.

233 – масала

Алгоритм «Кублар йиғиндиси».

$M=0$ ни оламыз $i-1$, j эса биринчи сон, унинг учун $j^3+1 \geq N$. $K=i^3+j^3$ деб оламыз ва « K » ни « N » билан таққослаймиз. Уч ҳол бўлиши мумкин:

- (1) агар $k < N$ бўлса, $i=i+1$ ни ҳосил қиламиз.
- (2) агар $K > N$ бўлса, $j=j-1$ ни ҳосил қиламиз.
- (3) агар $K=N$ бўлса, $M=M+1; j=j-1; i=i+1$ ни ҳосил қиламиз.

Бу ҳисоблашлар $i \leq j$ бўлгунча давом эттирилади. Шунингдек, (3) ҳолда $i=i+1$ ўрнига дарров $i=i+2$ ёзиш қулайлигини билдирамыз.

Бошида $i \leq j \leq \sqrt[3]{N} + 1$ бўлгани ва ҳар бир кадамда i ўсгани ёки j камайгани учун кадамлар $\sqrt[3]{N}$ дан катта бўлмайди.

Алгоритм тўғрилигини исботлаш учун $P(i,j)=\{(x,y)\}$ орқали $i \leq x \leq y \leq j$ ва $x^3+y^3=N$ шартларни қаноатлантирувчи шундай x ва y натурал сон жуфтлари тўпламини белгилаймиз.

Фараз қиламиз, $k=i^3+j^3$ бўлсин, агар $k < N$ бўлса, $P(i,j)$ тўпландаги ихтиёрий (x,y) жуфтлик учун $i^3+j^3 < x^3+y^3$ бўлади, лекин $y \leq j$, бундан $i < x$ ва, демак, агар $k < N$ бўлса, $P(i,j)=P(i+1,j)$.

Худди шунга ўхшаш, агар $k > N$ бўлса, $P(i,j)=P(i,j-1)$, ниҳоят, $k=N$ да, (i,j) жуфтлик $P(i,j)$ дан олиб ташланганидан кейин, агар $k=N$ бўлса, $P(i,j) \setminus (i,j)=P(i+1, j-1)$ бўлади.

Дастур

```

Program Kublar_yig'indisi;

```

```

var

```

```

    i,j,m,k,n    : integer;

```

```

begin

```

```

    writeln ('N:= ');

```

```

    readln (n);

```

```

    m:=0; j:=1; i:=1;

```

```

while j*j*j+1<n do j:=j+1;
repeat
    k:=i*i*i+j*j*j;
    if k=n then m:=m+1;
    if k<=n then i:=i+1;
    if k>=n then j:=j-1;
until i>j;
writeln (m);
end.

```

234 – масала

Алгоритм «Оддий бўлувчилар».

$i=2,3,\dots$ сонларини i сони N сонининг бўлувчиси бўлгунча алмаштираемиз. N ни i га бўламиз

$$N = N / i$$

ва бўлишни N_i сони i га бўлингунча такрорлаймиз. Шундан кейин, $N > 1$ бўлса, i нинг навбатдаги қийматига ўтамиз ва ҳоказо. Шундай тарзда бошланғич N сонининг ҳамма оддий бўлувчиларини ва фақат уларни топиш мумкинлигини сезиш қийин эмас.

Бошда 0 га, кейинчалик, охириги бўлувчига тенг бўлган, яна битта j ўзгарувчини киритамиз. Янги бўлувчи биринчи марта j дан фарқ қилади ва чоп этилади. Дастур ишини тезлаштириш учун, $i = 2$ ҳолини алоҳида кўриш ва кейинчалик эса i нинг тоқ қийматлари бўйича силжиш мумкин.

Дастур

```

Program Oddiy_bo'luvchilar;
var
    i,j,n : integer;
begin
    writeln ('N:= ');
    readln (n);
    writeln ('program get N:= ');
    j:=0; i:=2;
    while n>1 do
        begin

```

```

        while (n mod i) <> 0 do i:=i+1;
        if i <> j then
            begin
                writeln(i);
                j:=i;
            end;
        n:=n div i ;
    end;
end.

```

235 – масала

Алгоритм «Кўпхад».

$P_0(x)=1$ деб олиб, $P_m(x)=a(m,0)+a(m,1)x+\dots+a(m,m-1)x^{m-1}+x^m$ кўпхадни аниқловчи, дастлабки m та қавс кўпайтувчиларини $P_m(x)$ деб белгилаб олиш қулай:

$$P_m(x) = (x-x[1]) * \dots * (x-x[m])$$

$$P_m(x) = P_{m-1}(x)(x-x[m])$$
 эканлиги келиб чиқади ва $a(m,i)$

коэффициентлар $a(m-1,i)$ орқали қуйидаги тартибда ифодаланади:

$$a(m,m-1) = a(m-1,m-2) - x[m]$$

$$a(m,i) = a(m-1,i-1) - a(m-1,i)x[m]$$

$$a(m,0) = -a(m-1,0)x[m]$$

Агар $a(m,i)$ коэффициент A массивнинг $a[i]$ элементига жойлаштирилса, унда кўпхаднинг P_{m-1} коэффициентларидан, P_m кўпхаднинг коэффициентларига ўтиш ($m>1$ да) қуйидаги операторлар орқали бажарилади:

$$a[m-1] = a[m-2] - x[m]$$

$$a[i] = a[i-1] - a[i]x[m], \quad i = m-2, \dots, 1 \text{ учун}$$

$$a[0] = -a[0] - [m]$$

Бу иш таклиф қилинган дастурда амалга оширилган. У бошланғич ва сўнгги қийматларга тузатишлар билан иккиламчи ташқи (m бўйича 1 дан n гача) ва ички (i бўйича $m-1$ дан 0 гача) циклга эга..

Дастурни бошқача қилиб ҳам, ташқи циклни i бўйича $n-1$ дан 0 гача ва ичкисини $n-i$ дан n гача, расмийлаштириш мумкин эди.

```

Program Ko'phad;
const NN=100;
var
    i,m,n : integer;
    x : array [1..NN] of real;
    A : array [1..NN-1] of real;
begin
    Writeln ('N:=');
    readln (n);
    for i:=1 to n do
        begin
            writeln ('x[' ,i,']:=');
            readln (x[i]);
        end;
    a[0]:=-x[1]; a[1]:=1;
    for m:=2 to n do
        begin
            a[m-1]:=a[m-2]-x[m];
            i:=m-2;
            while i>=1 do
                begin
                    a[i]:=a[i-1]-a[i]*x[m];
                    i:=i-1;
                end;
            a[0]:=-a[0]*x[m];
        end;
        writeln;
        for i:=0 to n-1 do writeln ('a[' ,i,']:=',a[i]);
end.

```

236 – масала

Алгоритм «Қисқа кўпайтувчилар».

$A[n]$ массивда «қисқа кўпайтувчи»ларни ҳосил қилиб ва ўсиб бориш тартибида жойлаштириб борамиз. Дастлабки биринчи i қисқа кўпайтувчилар ҳосил қилинган бўлсин:

$$a[1] < \dots < a[i] \quad (*)$$

Ва (*) га кирувчи сонларнинг k_2 ва k_3 (улар учун $a_2 = 2 * a[k_2] + 1$ ва $a_3 = 3 * a[k_3] + 1$ тенгликлар бажарилади) минимал рақамлари маълум бўлсин, k_2, k_3 лар (*) га кирмайди.

(*) га яна битта ҳад киритилганда нима бўлишини кўрамиз. Маълумки,

$$a[i+1] = \min(a_2, a_3)$$

Агар $a[i+1] = a_2$ бўлса, k_2 1 га, агар $a[i+1] = a_3$ бўлса k_3 1 га (агар $a[i+1] = a_2 = a_3$ бўлса, иккала k_2 ва k_3 сонлар 1га) ўсади. Аввал $i=1$ $a[1]=1$, $k_2=1$, $k_3=1$. Навбатдаги $a[2]$, $a[3]$, ... ҳадлар баён этилган алгоритм билан ҳисобланади.

Дастур

```
Program Qisqa_ko'paytuvchilar;
```

```
const NN=100;
```

```
var
```

```
  i,k2,k3,a2,a3,n : integer;
```

```
  A : array [1..NN] of integer;
```

```
begin
```

```
  writeln ('N:=');
```

```
  readln (n);
```

```
  k2:=1; k3:=1; a[1]:=1;
```

```
  writeln;
```

```
  for i:=2 to n do
```

```
  begin
```

```
    a2:=2*a[k2]+1;
```

```
    a3:=3*a[k3]+1;
```

```
    if a2<=a3 then
```

```
      begin
```

```

        a[i]:=a2;
        k2:=k2+1;
    end;
    if a3<=a2 then
        begin
            a[i]:=a3;
            k3:=k3+1;
        end;
    writeln ('a[i]:=');
end;
end.

```

237 – масала

Алгоритм «Покер»

Масаланинг оддий ечими. Берилган A массивда нечта тенг жуфтликлар борлиги саналади. Уларнинг сони комбинацияни бир қийматли тавсифлар экан. Тенг жуфтлар сонини ҳисоблаш учун S ҳисобчини нолга тенглаштирамиз, $a[i]$ нинг ҳар бир элементини $a[i]$ нинг кейинги қолганлари билан $j=i+1, i+2\dots$ учун таққослаймиз ва, $a[i]=a[j]$ бўлган ҳолда, S га 1 тадан қўшамиз.

Чоп этиш учун керакли P жавоб (биринчи иккита ҳолдан бошқасида, $P=7-S$) S сўчтчикнинг қиймати бўйича жадвалда ҳосил бўлади:

S	P	S	P
$4+3+2+1=10$	1	$1+1 =2$	5
$3+2+1 =6$	2	$1 =1$	6
$2+1+1 =4$	3	$0 =0$	7
$2+1 =3$	4		

Дастур

```

Program Poker;
var
    i,j,p :integer;
    A : array [1..5] of integer;
begin
    p:=0;

```

```

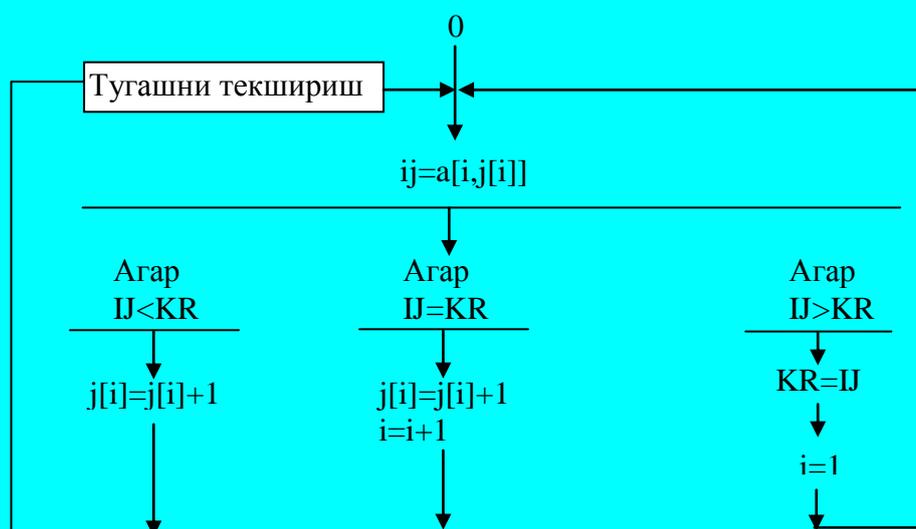
for i:=1 to 5 do readln (A[i]);
writeln;
p:=0;
for i:=1 to 4 do
  for j:=i+1 to 5 do
    if A[i]=A[j] then p:=p+1;
    if P=10 then writeln(1)
      else
        if P=6 then writeln (2) else writeln (7-P);
end.

```

238 – масала

Алгоритм «Тенг элементлар».

Ҳар бир i сатр учун $J[i]$ да, бу сатрда бориб етилган устун рақамини эслаб қолиш мақсадида, $J[1:m]$ массив ташкил қилиш ва бошда ҳамма $j[i]=1$ деб олиш керак. Тенг элементларга номзодларни KR деб, A массивнинг навбатдаги қўриладиган элементини эса IJ деб белгилаймиз. Аввал яна $KR = a[1,1]$ ва $i=1$ деб оламиз. Кейинги (камдан-кам учрайдиган) ҳолни блок-схема ёрдамида баён этиш мумкин:



$IJ = A[i, j[i]]$ ўзлаштиришдан олдин тугатишни текширишни қўшиб қўямиз. Агар $i > m$ бўлса, керакли қиймат топилган. Агар $j(i) > n$ бўлса, бу қиймат йўқ.

Лекин ёрдамчи $j[1:n]$ устун ўрнига ёрдамчи $io[1:n]$ сатрни ташкил қилиш мумкин. Аввал унга A массивнинг биринчи сатри киритилади, io сатрнинг no «ҳаракатдаги» элементлар сони « n »га тенг деб олинади.

Кейин массивнинг навбатдаги $i=1,2,3,\dots,m$ сатри **io** сатр билан таққосланади. **io** сатрда фақат **i** сатрда учраган элементлар қолдирилади (ва бошига сурилади), **io** да қолган **no** элементлар сони эса янги қиймат қабул қилади. Агар **no** қандайдир бир қадамда нолга айланса, тенг элементлар йўқ, акс ҳолда изланаётган элемент $io[1]$ бўлади. Бу алгоритм таклиф этилган дастурда амалга оширилган. Дастурда $io[j]$ элементлар ўрнига $a[o,j]$ элементлар олинган.

Дастур

```

Program Teng_elementlar;
const NN=20;
        MM=20
label 1,2;
var
        m,n,i,j,jo,no,jn    : integer;
        A : array [1..MM,1..NN] of integer;
begin
        writeln ('m,n:=');
        readln (m,n);
        for i:=1 to m do
        for j:=1 to n do
        begin
            writeln ('A['i,'o',j,']:=');
            readln (A[i,j]);
        end;
        for i:=1 to n do A[o,j]:=A[1,j];
        no:=n;
        for i:=2 to m do
        begin
            j:=1; jo:=1; jn:=1;
            1: if A[o,jo]<A[i,j] then jo:=jo+1
                else
                if (a[o,jo]>a[i,j]) then j:=i+1
                else
        begin
            a[o,jn]:=a[o,jo];

```

```

        j:=j+1;
        jn:=jn+1;
        jo:=jo+1;
    end;
    if (jo<no) and (j<=n) then goto 1;
no:=jn-1;
    if j<=n then goto 1;
    if no=0 then goto 2;
end;
writeln (a[0,1]); exit;
2: writeln ('Тенг элементлар йўқ');

```

end.

239 – масала Алгоритм «Йўл».

i_1, i_2, \dots, i_n сонларнинг ҳамма вариантларини кўриб чиқиш мумкин эмас, уларнинг сони m^n та. Лекин масалани ечишнинг бошқа, $k=1, 2, \dots, n-1$ бўйича индукция йўли мавжуд. k ва i_{k+1} нинг белгиланган сонларида $V[k, i_{k+1}] = \min(A[i_1, i_2, 1] + \dots + A[i_k, i_{k+1}, k])$ ни қўямиз, бу ерда \min мумкин бўлган i_1, i_2, \dots, i_k тўпламлардан олинади. Унда:

$V[1, i_2] = \min(A[i_1, i_2, 1])$ ҳамма i_1 лар бўйича,

$V[2, i_3] = \min(V[1, i_2] + A[i_2, i_3, 2])$ ҳамма i_2 лар бўйича,

$V[n-1, i_n] = \min(V[n-2, i_{n-1}] + A[i_{n-1}, i_n, n-1])$ ҳамма i_{n-1} лар бўйича ва изланаётган $R = \min V[n-1, i_n]$ ҳамма i_n лар бўйича эканлигини кўриш осон. Шундай қилиб, битта $V[k, i_{k+1}]$ ни ҳисоблаш учун фақат m та вариант (i_k танловнинг) қаралади. Белгиланган k ва ҳамма $V[k, i_{k+1}]$ лар учун m^2 вариантлар кўриб чиқилади, масаланинг ҳаммаси эса $m^2 * n$ дан кам бўлмаган вариантларни кўриб чиқишни талаб қилади.

Дастур

```

Program Yo'l;
const mm=10; nn=10;
var
    k,i,n,j,m : integer;
    x,r       : real;

```

```

A : array [1..mm,1..mm,1..nn] of real;
B,C : array [1..mm] of real;
begin
  writeln ('m,n=');
  readln (m,n);
  for i:=1 to m do
    begin
      b[i]:=0;
      for j:=1 to m do
        for k:=1 to n-1 do
          begin
            writeln ('a[',i,',',j,',',k,']=');
            readln (a[i,j,k]);
          end;
        end;
      end;
      for k:=1 to n-1 do
        begin
          for j:=1 to m do
            begin
              r:=b[i]+ a[i,j,k];
              for i:=2 to m do
                begin
                  x:=b[i] + a[i,j,k];
                  if x<r then r:=x;
                end;
              end;
              c[j]:= r;
            end;
          for j:=1 to m do b[j]:=c[j];
          end;
        r:=b[1]; for i:=2 to m do
        if b[i] <r then r:=b[i];
        writeln (r);
      end.

```

Алгоритм «Учбурчак ва нуқта».

Қуйидагини айтиб ўтамиз. $M(x,y)$ нуқта $M_1(x_1,y_1)$ ва $M(x_2,y_2)$ нуқтадан ўтувчи L_{12} тўғри чизиқда ётсин. Унда учбурчакларнинг ўхшашлигидан қуйидагини ҳосил қиламиз:

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

Махражларнинг нолга айланиши билан боғлиқ бўлган «қопқон»ни четлаб ўтиб, бу тенгламани қуйидаги кўринишда ёзамиз:

$$F_{12}(x,y) = (x-x_1) * (y_2-y_1) - (x_2-x_1) * (y-y_1) = 0$$

L_{12} тўғри чизиқда ётмаган $M(x,y)$ нуқталар учун $F_{12}(x,y)$ функциянинг нолга айланмаслиги аниқ. Бундан ташқари L_{12} тўғри чизиқдан чап томонда ётган (агар M_1 дан M_2 йўналишда қаралса) M нуқталар учун $F_{12}(x,y) < 0$ ва L_{12} дан ўнг томондаги M нуқталар учун $F_{12}(x,y) > 0$ эканлигини текшириш қийин эмас. Лекин $L_{12} = (M_1 - (-M_2))$, $L_{23} = (M_2 - (-M_3))$ ва $L_{31} = (M_3 - (-M_1))$ тўғри чизиқлар учун ҳосил қилинган учта соннинг: $F_{12}(x,y)$, $F_{23}(x,y)$, $F_{31}(x,y)$ ҳаммаси бир хил ишорага эга бўлса, M нуқта (M_1, M_2, M_3) учбурчак ичида шу вақтда ва фақат шу вақтда ётишини билдиради. Энди масалани ечиш учун M_1, M_2, M_3 нуқталарнинг координатаси учун $x[1:3]$ ва $y[1:3]$ иккита массивни киритиш ва $J(i,j) = \text{sign}[(X-X[i]) * (Y[j]-Y[i]) - (X[j]-X[i]) * (Y-Y[i])]$ функцияни аниқлаш кулай.

Унга уч марта мурожаат этиб, қуйидаги сонларни ҳосил қиламиз:

$$t_1 = \Phi(1,2), t_2 = \Phi(2,3), t_3 = \Phi(3,1)$$

Агар улар тенг бўлса, M нуқта учбурчак ичида, тенг бўлмаса, учбурчак ташқарисида (ёки чегарада) ётади.

Агар дастурлаш тили ҳам, массив ҳам ўзгарувчини битта ҳарф билан белгилашга имкон бермаса, M нуқта координаталарини $x[0]$, $y[0]$ ёки x_t , y_t ва х.к. шунга ўхшаб белгилаш мумкин.

Бу масаланинг бошқа ечими юзаларни таққослашга асосланган. Берилган (M_1, M_2, M_3) учбурчак юзи S билан, M нуқтани M_1, M_2, M_3 учларнинг иккитаси билан туташтирганда ҳосил бўладиган учбурчаклар юзалари эса S_1, S_2 ва S_3 билан белгиланади. Агар $S = S_1 + S_2 + S_3$ бўлса, нуқта учбурчак ичкарида, акс ҳолда ташқарида ётган бўлади. Бу ечим таклиф этилган дастурда келтирилган.

Учбурчак юзи Герон формуласи билан ҳисобланади. Агар $S_1+S_2+S_3 > 1,000001 * S$ бўлса, нукта учбурчак ташқарисида ётган деб ҳисобланади. Ҳисоблаш хатолигини 1,000001 кўпайтирувчи ҳисобга олади.

Дастур

```
Program Uchburchak_va_nuqta;  
type Point= record;  
    x : real;  
    y : real;  
end;  
var  
    TR : array [1..3] of Point ;  
    t : Point;  
    i : 1..3;  
function Dis(p,g : Point) : real;  
begin  
    Dis:=sqrt(sqr(p.x-g.x)+sqr(p.y-g.y));  
end;  
function Grn(a,b,c:Point):real;  
var da,db,dc,p:real ;  
begin  
    da:=Dis(c,b); db:=Dis(a,c); dc:=Dis(a,b);  
    p:=(da+db+dc)/2;  
    Grn:=sqrt(p*(p-da)*(p-db)*(p-dc));  
end;  
begin  
    for i:=1 to 3 do  
        begin  
            writeln ('x',i,',',',y',i,'=');  
            readln (tr[i].x,tr[i].y);  
        end;  
    writeln ('x,y=');  
    readln (t.x,t.y);  
    if Grn (tr[1],tr[2],tr[3])*1.000001< Grn(t,tr[1],tr[2]+ Grn(t,tr[1],tr[3])+  
    Grn(t,tr[2],tr[3]) then writeln('йўқ')
```

else writeln ('ха');

end.

241 – масала

Алгоритм «Тартибланган касрлар».

Дастурни ёрдамчи массивларсиз ва касрлар қисқаришини текширмасдан тузиш мумкин. Бунинг учун m/n каср киритилади ва аввал $m=0$, $n=1$ деб ҳисобланади, m/n чоп этилади. Кейин ҳамма a/b касрлар ичидан m/n дан катталари ва $b \leq p$ ($p=7$ шартда) лардан энг кичиги танланади. Бу i/j бўлсин. Агар $i/j < 1$ бўлса, m/n каср i/j каср билан алмаштирилади ва жараён давом эттирилади.

Дастурнинг баъзи бир хусусиятларини кўрсатиб ўтамиз.

Берилган $b=2, \dots, p$ махражлар учун сурат танланмайди, балки $a=m*b/n+1$ формула ҳисобланади.

Ҳамма m/n га тенг бўлган касрлар ичидан, у энг кичик махражга эга эканлигидан, ҳар бир топилган m/n каср автоматик равишда қисқармайдиган бўлади.

Шунингдек, касрларни таққослашда хусусийларини (машинада тақрибан ифодаланадиганларини) эмас, балки кўпайтмаларини ($a/b < i/j$ эмасу, лекин $a*j < b*i$ бўлганларини) таққослаш яхшилигини таъкидлаймиз.

Дастур

Program Tartiblangan_kasrlar;

var

p, m, n, a, b, i, j :integer;

begin

writeln ('p=');

readln (p);

$m:=0$; $n:=1$;

repeat

writeln (m:4, '/', n:4, '=', m/n);

$i:=1$; $j:=1$;

for $b:=2$ **to** p **do**

begin

```

        a:=m*b div n+1;
        if a*j<b*i then
        begin
            i:=a; j:=b;
        end;
    end;
    m:=i; n:=j;
until i>=j;
end.

```

242 – масала

Алгоритм «Тўртликлар жуфти».

Бу масала катта бўлмаган «қоидага» эга, чунки керакли хоссали минимал сонлар ўрнига, қўшилувчиларни кўриб чиқишнинг маълум бир ташкил этилишида, учраши мумкин бўлган биринчи сонни ҳосил қилиш мумкин. Изланаётган соннинг икки хил ифодаланиши (квадратлар йиғиндиси кўринишида) энг катта қўшилувчи билан фарқланиши кераклиги ҳисобга олинса, дастур ишини яна ҳам тезлаштириши мумкин.

Дастур

```

Program To`rtliklar_jufti;
label 1,2;
var
    i,n,j,k,p : integer;
    i1,j1,k1,p1 :integer;
    b : boolean;
begin
    for n:=2 to max int do
    begin
        b:=false;
        for i:=1 to n trunc (sqrt(n)) do
        begin
            for j:=1 to i do
            begin
                if i*i+j*j>=n then goto 1;

```

```

    for k:=1 to j do
        if i*i+j+k*k<n then
            begin
                p:=trunc(sqrt(n-i*i-j*k*k));
                if (i*i+j*k*k+p*p=n) and (p<=k) then
                    begin
                        if (b) then goto 2;
                        i1:=i; j1:=j; k1:=k; p1:=p;
                        b:=true;
                    end;
                end;
            end;
        end;
    end;
1: end;
end;
2: writeln (i*i+j*k*k+p*p, '=');
   writeln (I, ', ', I, '+', j, ', ', j, '+', k, ', ', k, '+', p, ', ', p, '=');
   writeln (i1, ', ', i1, '+', j1, ', ', j1, '+', k1, ', ', k1, '+', p1, ', ', p1, '=');
end.

```

243 – масала

Алгоритм «M+1 сонни иккиликда ёзиш».

a_0, a_1, \dots , сонлар кўздан кечирилади. Биринчи учраган нолгача бирлар ноллар билан алмаштириб борилади. Биринчи нол учрагач, алмаштириш тўхтатилади. Фақат, ечим шартдаги каби «n» та эмас, балки n+1 та сонга эга бўлиши мумкинлигини ҳисобга олиш керак.

Дастур

Program Sonni_ikkilikda_yozish;

var

 i, j, n : integer;

 b : boolean;

begin

 writeln ('n:=', n);

 readln (n);

 b:=true;

for i:=1 **to** n **do**

```

begin
  readln(j);
  if b then writeln (' ',1-j) else writeln (' ',j);
  if j:=0 then b:=false;
end;
if b then writeln ('1');
end.

```

244 – масала

Алгоритм «Квадратлар йиғиндиси».

Дастур осон ўқилади.

Бу масала «квадрат илдиз» функцияси қўлланилмаганида ҳам ечимга эга. Унинг ечилишини, текширилатган элементни олдингиси билан эмас, кейингиси билан, биринчи такрорлашгача таққослаб, тезлаштириш мумкин. Бундай таққослашлар сони $m*m$ дан $m*k$ гача камаяди, бу ерда m -хамма сонлар миқдори, k -берилган массивдаги ҳар хил сонлар миқдори.

Дастур

```

Program Kvadrat_summasi;
var
  i,j,m : integer;
  ff     : boolean;
begin
  readln (m);
  write('m:=',m);
  i:=1;
  ff:=true;
  while (2*i*i<=m) and ff do
    begin
      j:=round (sqrt(m-i*i));
      if (i*i+j*j=m) then ff:=false
      else i:=i+1
    end;
  if ff then writeln('йўқ')
  else writeln(i,'x',i,'+',j,'x',j,'='m);
end.

```

245 – масала

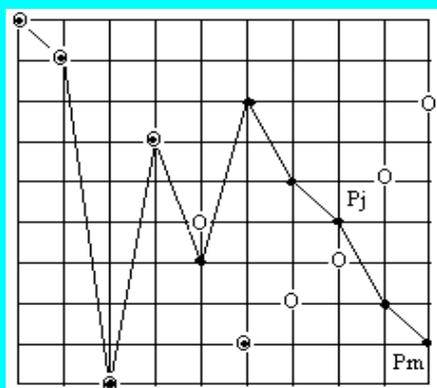
Алгоритм «Ўрин алмаштириш».

Бу масаланинг хусусияти ҳамма ўрин алмаштиришларни ҳосил қилишдадир. Бу билан у умумий танлаш масаласидан фарқ қилади ва масаланинг ечилишини енгиллаштиради. Шунга қарамай, масаланинг баёни дастурдан узунроқдир.

1 дан « m » гача бўлган сонларнинг барча ўрин алмаштиришларини кўз олдимизга келтирамиз. Хаёлан, уларни луғат тартибида жойлаштирамиз ва амалда эса берилган ўрин алмаштириш бўйича бевосита қуйидагини ҳосил қилишни ўрганамиз. Шунда биринчи ўрин алмаштиришдан $(1, 2, \dots, m)$, биз кетма-кет бу сонларнинг барча ўрин алмаштиришларини ҳосил қиламиз. Йўл-йўлакай $1, 2, \dots, m$ сонларнинг ҳар бир $P=(P_1, P_2, \dots, P_m)$ ўрин алмаштириши учун берилган A сонларнинг худди шундай $(A(P_1), A(P_2), \dots, A(P_m))$ ўрин алмаштиришини чоп этамиз.

$1, 2, \dots, m$ сонларнинг берилган $P=(P_1, P_2, \dots, P_m)$ ўрин алмаштириши бўйича, бевосита навбатдагисини ҳосил қилиш учун, P_1, P_2, \dots, P_m сонлар охиридан қараб чиқиш керак. Ўнг томондаги ҳадидан кичик бўлган $(P_i < P_{2i+1})$, биринчи учраган P_i ҳадда тўхталинади. Агар бундай ҳад бўлмаса, P ўрин алмаштириш $(m, m-1, \dots, 1)$ кўринишга эга (яъни охиригиси) бўлади. $P_{i+1} > P_{i+2} > \dots > P_m$ ҳадлар камаювчи кетма-кетликни ҳосил қилиши аниқ. Уларнинг ичидан P_i дан катта бўлган (агар охиридан қаралса) биринчи P_j ҳадни топамиз ва уларнинг ўрнини алмаштирамиз.

$P_{i+1}, P_{i+2}, \dots, P_m$ ҳадларни ўсиб бориш тартибида алмаштириб қўйиш қолади ва изланган (уни $Q=(g_1, \dots, g_m)$ деб атаймиз) ўрин алмаштириш ҳосил бўлади. (4-расм).



4-расм

- - P ўрин алмаштириш
- - Q ўрин алмаштириш

Ҳақиқатдан ҳам, $P > Q$, уларнинг биринчи $i-1$ ҳадлари мос тушади, P_i эса g_i дан кичик ($P_i < g_i$) бўлади (чунки ҳадни танлашнинг ўзидан $g_i = P_i <$

P_j). Кейин, ўзининг берилган биринчи i ҳадларида P - максимал, Q - минимал ўрин алмаштириш бўлади, чунки R да қолган ҳадлар камайиш, Q да эса ўсиш тартибида келади. Ниҳоят, R ўрин алмаштириш P ва Q ўртасида ётган бўлса, унинг дастлабки $i-1$ ҳадлари P ва Q нинг дастлабки ҳадлари билан мос тушади, r_i ҳад эса p_i га ёки g_i га тенг бўлади, чунки биринчи $i-1$ банд ҳадларда P_i ва g_i ўртасида жойлашган сонлар бўлмайди. Агар $p_i=r_i$ ва $P \leq R$ бўлса, $P=R$ (чунки P_1, P_2, \dots, P_i берилганларда P максимал), агар $r_i=g_i$ бўлса, худди шундай $R=Q$ бўлади.

Дастурда бу қуйидагича амалга оширилади. Жорий ўрин алмаштириш учун P массив ҳосил қилинади, у биринчи ўрин алмаштириш $P=(1,2,\dots,m)$ билан тўлдирилади ва берилган A -массив ҳадларининг тегишли ўрин алмаштирилиши чоп этилади.

Навбатдаги P ўрин алмаштириш ҳосил қилинган ва унга мос A ўрин алмаштириш чоп этилган бўлсин. i максимал рақамга эга бўлган $P_i < P_{i+1}$ элемент изланади. Агар у бўлмаса, P ўрин алмаштириш охиригиси бўлади. Акс ҳолда $P_i < P_j$ шарт бажариладиган $j < i$ энг катта рақам изланади. $P_i < P_j$ элементларнинг ўрни алмаштирилади, шундан кейин $P_{i+1}, P_{i+2}, \dots, P_m$ кетма-кетликда тартиб тескарисига ўзгаради: бунинг учун P_{i+1} ва P_m , кейин эса P_{i+2} ва P_{m-1} , ва ҳоказо лар ўринларини алмаштиришади.

Навбатдаги ўрин алмаштиришни ҳосил қилиш шу билан тугайди ва A массивнинг тегишли ўрин алалмаштирилиши чоп этилади.

Дастур

```

Program O`rin_almashtirish;
const mm=100;
var
    m,i,j,k,n : integer;
    A,P : array [1..mm] of integer;
begin
    readln (m);
    write ('m=',m);
    for i:=1 to m do
        begin
            readln (a[i]);
            p[i]:=i
        end;

```

```

for i:=1 to m do write ( a[i], ' ');
writeln;
for i:=m-1 downto 1 do if (p[i]<p[i+1]) then
  begin
    n:=p[i];
    for j:=m downto i do
      if ( n<p[j] ) then
        begin
          p[i]:=p[j]; p[j]:=n;
          k:=1;
          while i+k<m-k+1 do
            begin
              n:=p[i+k];
              p[i+k]:=p[m+1-k];
              p[m+1-k]:=n;
              k:=k+1;
            end;
            j:= i;
          end;
        for i:=1 to m do write (a[p[i]], ' ');
        writeln;
      end;
    end.

```

246 – масала

Алгоритм «Туб сонлар».

Топилган туб сонлар учун ҳисоблашни тезлаштириш мақсадида жадвал ҳосил қилиш фойдали. Навбатдаги номзодни фақат шу жадвалдаги сонларга бўлинишини текшириш керак. Жуфт сонлар, табиий, қаралмайди. Жадвал камида $\sqrt{M}/2$ сонлар учун ишлатилиши мумкин. Яъни 1000 та сонли жадвал 4000000 гача бўлган сонларни чоп этиш учун етарли.

Алгоритм ва дастурни, 3 га бўлинадиган сонларни эътибордан четда қолдириб яхшилаш мумкин.

```

Program Tub_son;
const n=200;
label 1,2;
var
    m,i,j,k,q : integer;
    P : array [1..n] of integer;
begin
    write ('M=');
    readln (m);
    write ('N=',N);
    it (m>=2) then writeln (2);
    if (m>=3) then writeln (3);
    k:=1; P[k]:=3; i:=5;
    while i<=m do
        begin
            for j:=1 to k do
                begin
                    g:=p[j];
                    if (g*g>i) then goto 1;
                    if (i mod g=0) then goto 2;
                end;
            if k=n then I=m-1
                else
            1: begin
                    writeln (i);
                    if k<=n-1 then
                        begin
                            k:=k+1;
                            p[k]:=I;
                        end;
                    end;
                2: i:=i+2;
            end;
        end;
    end.

```

247 – масала

Алгоритм «Тенг сонларни излаш».

Бу масalani қониқарли ҳал қилиш учун, $i=p$ ва $j=g$ ҳолларда, чалқашмаслик мақсадида, икки марта таққослаш учун бир хил $a[i,j]$ ва $a[p,g]$ элементлар жуфтини олмаслик керак.

Дастур

```
Program Teng_sonlar;
```

```
label 5;
```

```
var
```

```
  i,j,p,b,g : integer;
```

```
  ff       : boolean;
```

```
  A : array [1..2,1..15] of real;
```

```
begin
```

```
  for j:=1 to 15 do
```

```
    begin
```

```
      write ('A[1, ',j, ']:=');
```

```
      readln (a[1,j]);
```

```
      writeln ('A[2, ',j, ']:=');
```

```
      readln (a[2,j]);
```

```
    end;
```

```
  ff:=false;
```

```
  for i:=1 to 2 do
```

```
    for j:=1 to 15 do
```

```
      for p:=1 to 2 do
```

```
        begin
```

```
          if i<p then b:=1 else b:=j+1;
```

```
          for g:=b to 15 do
```

```
            if (A[i,j]=A[p,g] ) then
```

```
              begin
```

```
                ff:=true;
```

```
                goto 5;
```

```
              end;
```

```
          5: if ff then writeln ('A[',i, ', ',j, ']=A[',p, ', ',g, ']);
```

```
            else writeln ('__');
```

end

end.

248 – масала

Алгоритм «Рақамларнинг берилган йиғиндиси»

Изланаётган соннинг рақамларига кўра учламчи циклини ёзиш мумкин:

i бўйича 0 дан 9 гача, j бўйича 0 дан 9 гача, k бўйича 1 дан 9 гача.

Унда $i+j+k$ йиғиндисини ҳисоблаш ва u берилган сонга тенг бўлса, уч хонали: $M = i + 10j + 100k$ сонни чоп этиш керак. Лекин бу яхши ечим эмас. Унда мумкин бўлган 100 та цикл ўрнига 900 та цикл бажарилади.

Қабул қилиниши керак бўлган ечим i ва j бўйича иккиламчи циклга эга, K эса « n » нинг берилган йиғиндиси билан ҳисобланади.

$$K = n - i - j$$

$1 \leq k \leq 9$ текширишни киритиб, таклиф этилган дастурга келиш мумкин. Дастурни $n \leq 27$ шартни текшириш билан бошлаб, $n < 18$ ҳолида ҳисоблашнинг бошқа тезланишларини киритиб, дастурни такомиллаштириш мумкин.

Дастур

```
Program Raqam_beril_summasi;
```

```
var
```

```
    i,j,k,n : integer;
```

```
begin
```

```
    writeln ('n:= ',n);
```

```
    readln (n);
```

```
    if n in [1..27] then
```

```
        for i:=0 to 9 do
```

```
            begin
```

```
                for j:=0 to 9 do k:=n-i-j;
```

```
                if k in [1..9] then writeln (i+10*j+100*k);
```

```
            end
```

```
end.
```

249 – масала

Алгоритм «Функция».

$f(n)$ формулани топиш қийинлигидан ва n аргументининг ҳар бир камайиш қадамида, кўздан кечирилиш керак бўлган, функциялар сони ўсаётгандай бўлиб кўринишидан, масалани ечиш мумкин эмасдай бўлиб туюлади. Ҳақиқатдан ҳам, $f(2n+1)$ функциядан чиқилса, иккита функция ҳосил қилинади:

$$f(2n+1) = f(n) + f(n+1)$$

Иккита n ва $n+1$ аргументлардан биттаси тоқ. Кейинги қадамда у яна иккита функцияни туғдиради ва ҳоказо. Лекин иккинчи қадамни бажара туриб, бу функция аввалгидай учта эмас, балки иккита ва ҳамма вақт иккита бўлиб қолаётганини сезиш мумкин.

Дастурни ҳосил қилувчи бу ҳолни исботлаш учун, берилаётган бир аргументли $f(n)$ функция билан бир қаторда қуйидаги n,i,j -уч функция киритилади:

$$g(n,i,j) = if(n) + if(n+1)$$

Унинг учун рекуррент формулалар: $g(2n,i,j) = g(n,i+j,j)$, $g(2n+1,i,j) = g(n,i,i+j)$ осон текширилади. Энди изланаётган $f(n)$ қийматни қуйидагича ёзиш мумкин:

$$f(n) = g(n,1,0),$$

рекуррент формулаларини кўп марта қўллаб эса, g функциянинг биринчи аргументини нол қилиш ва $f(n) = g(n,1,0) = \dots f(n) = g(0,i,j) = j$ ни ҳосил қилиш мумкин. Дастурни ёзишда тил талаб қилаётган расмийлаштириш ишларини бажариш қолди, холос.

Дастур

Program Funksiya;

var

i,j,n : integer;

begin

readln (n);

writeln ('n:= ',n);

i:=1; j:=0;

while n>0 **do**

begin

```

        if mod 2=0 then i:=i+j else j:=j+i;
        n:=n div 2;
    end;
writeln(j);

```

end.

250 – масала

Алгоритм «Тўртбурчаклар».

Бу масала «бир фикрлаш»да ечилади: тўртбурчакларнинг шимолий-ғарбий бурчаклари (бошқача айтганда-юқори чап) қанча бўлса, тўртбурчаклар шунча. Фақат бурчак чегарада бўлган ҳолда чалкашилмаса бўлгани. Бу қийинчилик дастурларда ҳар хил ҳал қилиниши мумкин. Фақат дастурда $i > 1$ and $a[i-1,j] = 0$ турдаги ифодани ишлатмаслик кераклигини огоҳлантирамиз. Чунки, агар индексация бирдан бошланса, $i=1$ да бундай ифода синтактик хато бўлиб ҳисобланади.

Дастур

```

Program To`rtburchak;
const nn=100; mm=100;
var
    i,n,s,j : integer;
    a : array [0..mm,0..nn] of byte;
begin
    writeln ('m,n=');
    readln (m,n);
    for i:=0 to m do
        for j:=0 to n do
            if (i=0) or (j=0) then a[i,j]:=0 else
                begin
                    writeln ('a[',i,',',j,']:=');
                    readln (a[i,j]);
                end;
            s:=0;
        for i:=1 to m do
            for j:=1 to n do

```

```

        if (a[i,j]=1) and (a[i-1,j]+a[i,j-1]=0) then s:=s+1;
    writeln(s);
end.

```

251 – масала
Алгоритм «Бит-реверс».

Аниқлик учун m ва $V(m)$ нинг бир нечта қийматларини ўнлик ва иккилик санок системаларида ёзамиз:

Система	10	2	2	10
m ва $V(m)$	512	1000000000	0000000001	1
қийматлари	513	1000000001	1000000001	513
	514	1000000010	0100000001	257
	515	1000000011	1100000001	769
	516	1000000100	0010000001	129

	1023	1111111111	1111111111	1023

Жадвалдан кўришиб турибдики, « m » сонининг бош иккилик разряди 512ини ифодалаябди ва унга, 1 миқдорини ифодаловчи, $V(m)$ сонининг кичик разряди мос келаябди. Кейинги разрядлар учун бу 256 ва 2. ва ҳоказо миқдорлар бўлади. Ҳамма вақт « m » сониди, $m \geq 512$ бўлгани учун бош разряднинг бир сони бор. Уни, « m »ни $m-512$ га алмаштириб, йўқ қиламиз ва бирни $V(m)$ сонига киритамиз.

Кейинги разрядда, агар янги қиймат $m \geq 256$ бўлса, « m »сониди бир бўлади. Агар у бўлса, уни йўқ қиламиз ва $V(m)$ га 2 сонини киритамиз. Навбатдаги бирнинг бор-йўқлигини текшираамиз ва ҳоказо. Иш « m » сони нолга айланганида тўхтатилади. Бу ишларни ўз ичига олган дастур ўзининг ҳисоблашлари учун $512 \cdot 10$ циклни сарфлайди.

$V(m)$ сонлари бўйича $V(m+1)$ сонини ясовчи дастур тезроқ ишлайди. $V(m)$ қийматларининг иккилик ёзувларини кўздан кечириб, қуйидагини тушуниб олиш мумкин: $V(m+1)$ ни ясашда $V(m)$ сонининг иккилик ёзувида чапдан ўнгга томон, бирларни ноллар билан биринчи нолгача алмаштира бориб (яъни $V(m)$ дан 512,256... ларни айира бориб), силжиш керак. Биринчи нолни бир билан алмаштириш керак, $V(m+1)$ сони ҳосил бўлади. Бу дастур, соннинг ярмини ҳисоблашга биттадан цикл, чорагини

ҳисоблашга иккитадан цикл, саккиздан бирини ҳисоблашга учтадан цикл ва ҳоказо, сарфлайди. Ҳаммаси бўлиб $512 \cdot 2$ тадан кам цикл кетади. Келтирилган дастур шунга асосан тузилган.

Дастур

```
Program Bit_revers;  
var  
    m,b,k : integer;  
begin  
    k:=512; m:=1;  
    writeln (m);  
    while m < 1024 do  
        begin  
            while m>= k do  
                begin  
                    m:=m-k; k:=k div 2;  
                end;  
            m:=m+k; k:=512;  
            writeln (m);  
        end  
end.
```

252 – масала

Алгоритм «Инверсия».

P массивни ноллар билан тозалаймиз. Навбатдаги $i = 1, 2, \dots, n$ ва T_i сонни оламиз. $T+1$ да нол элементлар учрагунча P_1, P_2, \dots элементлар бўйлаб ўтамиз ва уларнинг охиргисига i сонини ёзамиз.

Дастур

```
Program Inversiya;  
const nn=100;  
var  
    i,n,j,k : integer;  
    P,T : array [1..nn] of integer;
```

```

begin
  writeln ('n=',n);
  readln (n);
  for i:=1 to n do
    begin
      writeln ('T[',i,']=');
      readln (T[i]);
      P[i]:=0;
    end;
  for i:=1 to n do
    begin
      j:=0; k:=0;
      repeat
        k:=k+1;
        if P[k]:=0 then j:=j+1;
      until j>T[i];
    end;
    P[k]:=i;
  for i:=1 to n do
    writeln (P[i]);
end.

```

253 – масала

Алгоритм «Қўшилувчиларга ажратиш».

Қўшилувчилар ўрнини алмаштиришда ажратишларнинг такрорланмаслигига эришиш учун «n» сонининг шундай натурал қўшилувчиларга ажралишини кўриб ўтамизки, $n=m_1+m_2+\dots$, бунда, $m_1 \geq m_2 \geq \dots$

Ажралишни тартибга соламиз:

$$n=m'_1+m'_2+\dots \quad (m')$$

$$n=m''_1+m''_2+\dots \quad (m'')$$

- иккита ажратиш. Агар биринчи тенгмас $m'_i \neq m''_i$ жуфтдаёк, $m'_i > m''_i$, яъни ҳамма $j < i$ ва $m'_j > m''_j$ да $m'_j \neq m''_j$ бўлса (m') ажратиш (m'') ажратишдан олдин келади, деб ҳисоблаймиз. Шу қоидага кўра, $n=5$ сонини қўшилувчиларга ажратиш қуйидаги тартибда бўлади:

$5=5,$
 $5=4+1,$
 $5=3+2,$
 $5=3+1+1,$
 $5=2+2+1,$
 $5=2+1+1+1,$
 $5=1+1+1+1+1$

Берилган N сонининг қўшилувчиларга ажратишнинг ҳамма кўринишларини ҳосил қилиш учун ишни $n=n$ ажратишдан бошлаймиз. Агар навбатдаги ажратиш

$$n=m_1+m_2+\dots+m_i \quad (*)$$

ҳосил қилинган бўлса, уни чоп этамиз ва бевосита ундан кейин келувчисини тузишга ўтамиз. Бунинг учун массив бўйича ўнгдан чапга томон силжиб бориб, m_i, m_{i-1}, \dots, m_k ни, « m_k » нинг бирдан катта қўшилувчисигача, кўриб борамиз. Агар бунақаси учрамаса, ҳосил бўлган (*) ажратиш охиргиси ва иш тугаган, $m_k > 1$ қўшилувчи топилган бўлади. m_k ни бирга камайтириб ва ундан кейин келувчи бирлик қўшилувчиларни олиб ташлаб, бутун йиғиндини $S=1+i-k$ миқдорда камайтирамиз. Энди $j=k+1, k+2, \dots$, учун m_j нинг янги кетма-кет қўшилувчиларини аниқлаймиз. Агар $S > m_k$ бўлса, $m_j = m_k$ деб оламиз ва S ни m_k миқдорга камайтирамиз. Агар $S \leq m_k$ бўлса, $m_j = S$ деб оламиз ва шу билан n нинг қўшилувчиларга ажратишнинг янги кўринишини тугатамиз.

Дастур

```

Program Qo'shiluvchilarga_ajratish;
const NN=100;
label ra,br;
var
    n,i,k,t,s : integer;
    M : array [1..NN] of integer;
begin
    writeln ('n:=,');
    readln (n);
    writeln (n);
    m [1]:=n; k:=1; i:=1;

```

```

ra: t:=m[k]-1 ; s:=t+i-k+1;
  for i:=k to n do
    if s>t then
      begin
        m[i]:=t; s:=s-t
      end
    else
      begin
        m[i]:=s;
        goto br
      end;
  br: for k:=1 to 1 do writeln (m[k], ' ');
      writeln;
      for k:=i downto 1 do
        if m[k]>1 then goto ra
      end.
end.

```

254 – масала

Алгоритм «Ўнг катта».

Ечиш учун икки хил фикр ишлатилади. Биринчидан, $A[1:n]$ массив охиридан кўздан кечирилади; иккинчидан, A массив кейинги кўздан кечирилишида, унинг элементларини алмаштириш учун, керак бўладиган элементларидан, $B[1:n]$ массив ҳосил қилинади. Аниқроқ қилиб айтганда, $A[i]$ массивнинг навбатдаги элементи кўздан кечириляётганда, B массивда ҳамма $a[i+1, j-1]$ элементлардан катта $a[j](j>i)$ массив элементлари ётган бўлади. Элементлар A массивда қандай тартибда жойлашган бўлса, B массивда ҳам худди шундай тартибда жойлашган бўлади, фақат $B[k:n]$ кесмани тўлдира бориб ўнг томонга силжиган бўлади.

A ва B массивлар бир вақтда қайта ишланади. Аввал $k=n$: $b[k]=a[n]:a[n]=0$, деб олинади. Навбатдаги $a[i]$ элемент қаралаётганда, $A[i+1:n]$ соҳадаги ҳамма элементлар алмаштирилган ва $B[k:n]$ массив керакли тарзда тўлдирилган бўлади. $B[k:n]$ массивда биринчи $b[j]>a[i]$ элемент изланади. Агар шундай j рақам топилса, $k=j-1$; $b[k]=a[i]:a[i]=b[j]$, бажарилади. Агар, шундай j рақам бўлмаса, $k=n$: $b[k]=a[i]:a[i]=0$, деб олинади. Иккала ҳолда янги $B[k:n]$ кесма эълон қилинган ўз хоссаларини

сақлайди. Ҳар бир $a[i]$ элемент $b[j]$ элемент билан таққосланаётганда ёки $a[i]$ элемент B га киритилади ёки $b[j]$ элемент B дан йўқ қилинади. Шундай тарзда A массивнинг ҳар бир элементи биринчи турдаги (y B га киритилмаган ҳолда) битта таққослашни ва биттадан кўп бўлмаган – иккинчи турдаги (B дан йўқотилган ҳолда) таққослашни туғдиради. Демак, таққослашлар $2n$ дан, амаллар эса n тартибдан кўп эмас.

Дастур

```
Program O'ng_katta;
const NN=100;
label 1,2;
var
    j, i, k, n : integer;
    ff          : boolean;
    A,B : array [1..NN] of real;
begin
    writeln ('N:=');
    readln (n);
    for i:=1 to n do
        begin
            write ('A[',i,']:=');
            readln (A[i]);
        end;
    B[n]:=A[n]; A[n]:=0; k:=n;
    for i:=n-1 downto 1 do
        begin
            j:=k; ff:=false;
            while (j<=n) and not ff do
                begin
                    ff:=A[i]<B[i];
                    j:=j+1;
                end;
            if ff then
                k:=j-1-1;
                B[k]:=A[i];
                A[i]:=B[j-1];
```

```

end
    else
begin
    K:=n; B[k]:=A[i]; A[i]:=0;
end;
for i:=1 to n do writeln (A[i]);
end.

```

255 – масала

Алгоритм «Рюкзак».

30 кг дан оғир бўлган предметлар олиб ташлангач, қолганлари маълум бир тартибда жойлаштирилгач, вариантлар шажарасини қуйидагича аниқлаймиз. Навбатдаги $I=1,2,\dots,n$ йўлда i рақамли предметни қараймиз, i йўлнинг j вариантлари эса ҳамма вақт иккита бўлади: $j=0$ предметни олиш, $j=1$ предметни олмасликни билдиради. Тармоқлари n узунликка тенг иккиламчи дарахт ҳосил бўлади.

Берилган $A[1:n]$ ва $B[1:n]$ массивлардан ташқари $P[1:n]$ массив ва бир нечта ўзгарувчи киритамиз:

- i – навбатдаги предмет рақами,
- T – рюкзакдаги предметлар оғирлиги,
- Z – рюкзакдаги предметларнинг жами қиймати,
- ZM - кўрилган вариантларнинг максимал қиймати.
- $k \leq i$ предмет рюкзак олинса, $P[k]=0$
- $k \leq i$ предмет рюкзакка олинмаса, $P[k]=1$.

Бошда i, S, Z, ZM нолга тенглаштириб олинади. Вариантларни кўздан кечиришда предметнинг (ва унинг ҳамма давоми) қизиқиш туғдирмаслиги аниқ бўлиши билан кўриб чиқишни тўхтатиш муҳимдир. *Олдинга ҳаракат* қилишда (агар $S+A[i]<30$ бўлса), предметни рюкзакка қўйишга интиламиз. Бу ҳолда биз чап тармоқ бўйича борамиз:

$$S=S+A[i] : Z=Z+B[i] : P[i]=0$$

Агар предметни қўйиш мумкин бўлмаса, уни оламиз (яъни чапга кетувчи вариант тармоқларини ташлаб бориб, ўнг тармоқ бўйича ҳаракатланамиз) ва $P[i]=1$ деб оламиз. Иккала ҳолда ҳам энг охириги предмет кўрилмагунча олдинга ҳаракатни давом эттирамиз.

Агар ҳамма вариантлар кўздан кечирилган бўлса, вариант ҳосил қилинди. У ZM билан таққосланади.

if $ZM < Z$ then $ZM = Z$

ва *охирига* қараб ҳаракат бошланади.

Олдинга ҳаракат қилишда, олинган кетма-кет келувчи предметларнинг ҳамма гуруҳи ўтказиб юборилади (уларда $P[i]=0$), чунки битта шу гуруҳдаги ўзгариш рюкзакдаги предметларнинг жами қийматини туширади, холос. Кўрилган предметлар рюкзакдан йўл-йўлакай олиб ташланади:

if $P[i]=0$ then $S=S-A[i]$: $Z=Z-B[i]$

Шундан кейин, олдин олинмаган предметларнинг бутун гуруҳи ўтказиб юборилади (уларда $P[i]=1$), чунки бу гуруҳдаги ўзгариш олдин баҳоланиши керак бўлган чап тармоққа олиб келади.

Қисқа қилиб айтганда, биз $P[i]=0$ ва $P[i+1]=1$ ни ҳосил қиладиган шундай рақамга эришгунимизча, *охирига қараб ҳаракат қиламиз*. Бундай ҳаракатда рюкзакдан унда мавжуд бўлган предметлар олиб ташланади. Агар керакли i бўлмаса иш тугатилади.

Дастур

```
Program Ryukzak;  
const NN=100;  
      T=30;  
label 1;  
var  
      i,s,z,zm,n : integer;  
      A,B : array [1..NN] of real;  
      P : array [1..NN] of boolean;  
begin  
      writeln ('N:=');  
      readln (n);  
      for i:=1 to n do  
          begin  
              writeln ('A[',i,']:=');  
              readln (A[i]);  
              writeln ('B[',i,']:=');  
              readln (B[i]);  
          end;  
      S:=0; Z:=0; ZM:=0; i:=0;
```

```

1: for i:=i+1 to n do
    it S+A[i]>=T then P[i]:=false else
        begin
            S:=S+A[i];
            Z:=Z+B[i];
            P[i]:=TRUE
        end;
    if zm <z then zm:=z;
    for i:=n-1 downto 1 do
        begin
            if P[i+1] then
                begin
                    S:=S-A[i+1];
                    Z:=Z-B[i+1];
                end;
            if P[i] and not P[i+1] then
                begin
                    S:=S-A[i];
                    Z:=Z-B[i];
                    P[i]:=False;
                    Goto 1
                end;
        end;
    end;
writeln (ZM)
end.

```

Фойдаланилган адабиётлар рўйхати

1. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. – М.: Мир, 1989. – 360 с.: ил.
2. Зелковиц М., Шоу А., Гэннон Дж. Принципы разработки программного обеспечения: Пер. с англ. – М.: Мир, 1982. – 386 с.: ил.
3. Культин Н.Б. Программирование в Turbo Pascal 7.0. и Delphi / Второе издание, переработанное и дополненное. – СПб.: БХВ – Санкт-Петербург, 1999. – 416 с.: ил.
4. Культин Н.Б. Программирование на Object Pascal в Delphi 5. СПб.: БХВ – Санкт-Петербург, 1999. – 464 с.: ил.
5. Культин Н.Б. Turbo Pascal в задачах и примерах. – СПб.: БХВ – Санкт-Петербург, 2001. – 256 с.: ил.
6. Практическое руководство по программированию: Пер. с англ. Б.Мик, П.Хит, Н.Рашби и др.; под ред. Б.Мик, П.Хит, Н.Рашби. – М.: Радио и связь, 1986. – 168 с.: ил.
7. Фокс Дж. Программное обеспечение и его разработка: Пер. с англ. – М.: Мир, 1985. – 368 с.: ил.
8. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. 2-е издание. М.: Наука. –1990. –208 с.
9. Марченко А.Н. Программирование в среде Turbo Pascal 7.0. – К.: Век+, М.: ДЕСС, 1999 – 496 с., ил.
10. Фаронов В.В. Turbo Pascal 7.0. М.: Нолидж, -2000, -576 с., ил.
11. Файсман А.В. Профессиональное программирование на Турбо Паскале. Т.: ИнформЭкс – Корпорейшн, -1992. – 270 с., ил.
12. Очков В.Ф., Пухначев Ю.Ю. 128 советов начинающему программисту. – М.: 1991.
13. Липаев В.В. Проектирование программных средств. – М.: 1990.