

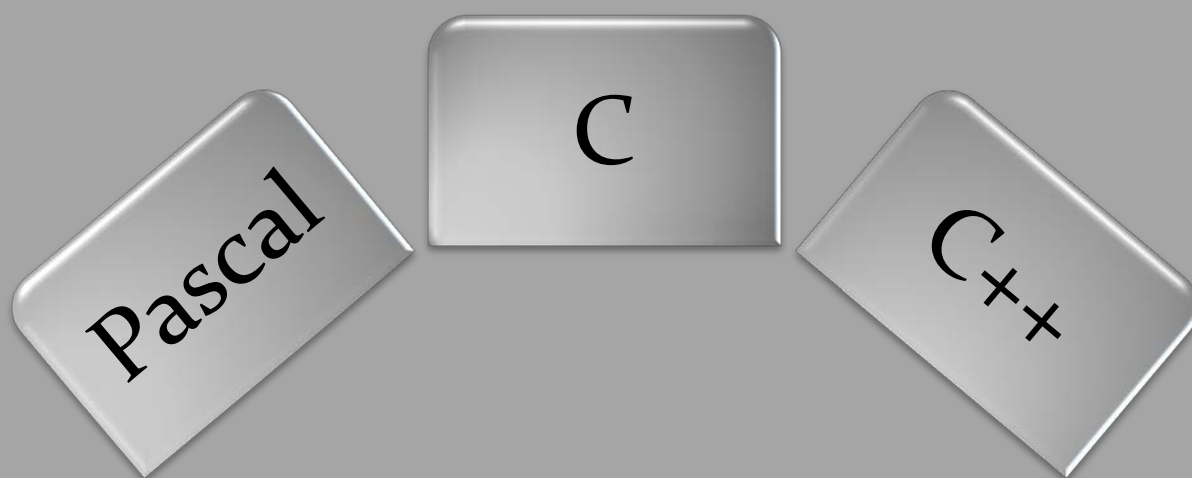
Grigore VASILACHE

Silviu GÎNCU

C U L E G E R E

DE PROBLEME LA INFORMATICĂ

pentru orele
de laborator



Capitolul 1

INSTRUCȚIUNI

§1 Operații de intrare/ieșire

<i>Operații de ieșire</i>	
Pascal	<pre>write(x); sau writeln(x); write(x1,x2,...xn); este echivalent cu write(x1); write(x2,); ...write(xn);</pre> <p>Parametrii din apel pot avea una din formele:</p> <p style="text-align: center;"> x $x:w$ $x:w:f$, unde x este o expresie de orice tip valoarea căreia se va afișa, w și f – expresii de tip integer – specificatori de format. Ca parametru pot fi și constante caracteriale, care se scriu în apostrofuri, și se afișează așa cum se scriu ca parametru. </p> <p>Fie următoarea secvență de program:</p> <pre>Var a:integer; b:real; ... a:=5; b:=7.236; writeln('Valoarea lui a este',a:6, ' b=',b:5:2)</pre> <p>va afișa:</p> <p>Valoarea lui a este 5 b= 7.24</p> <p>Atragem atenția că ultima cifră este 4 și nu 3 fiindcă a avut loc rotunjirea. Instrucțiunea <i>writeln</i> se deosebește de <i>write</i> prin faptul că după afișarea tuturor parametrilor se trece în linie nouă.</p>
C	<pre>printf(sir_de_ieșire,listă_de_ieșire);</pre> <p>unde :</p> <p><i>sir_de_ieșire</i> este un șir în componența căruia intră caractere ordinare și grupe de caractere speciale, ce se numesc formate sau specificații de conversie.</p> <p><i>listă_de_ieșire</i> reprezintă enumerarea elementelor de ieșire, ce se despart prin virgulă. În calitate de elemente pot fi constante, variabile și expresii.</p> <p><i>Exemplu :</i></p> <p>Fie date următoarele declarații :</p> <pre>int x=4,y=6; printf("%d+%d=%d",x,y,x+y);</pre> <p>În rezultatul execuției acestei secvențe de program va fi afișat mesajul 4+6 =10.</p> <pre>int a=24;</pre>

	<pre>double c=123.1234567;</pre> <p>instrucțiunea</p> <pre>printf("%d\n a=%5d\nc=%7.2g" ,a,a,c);</pre> <p>va afișa :</p> <pre>24 a= 24 numărul a este afișat pe 5 poziții c= 123.12 numărul c este afișat pe 7 poziții cu 2 cifre după virgulă</pre>
C++	<pre>cout<<flux1<<flux2<<...<<fluxn;</pre> <p>Fie date următoarele declarații :</p> <pre>int a=24; double c=123.1234567; cout<<a<<endl<<"a="<<setw(5)<<a<<endl; cout<<"c="<<setw(7)<<setprecision(2)<<c; cout << setfill ('x') << setw (10); cout << 77 << endl;</pre> <p>va afișa :</p> <pre>24 a= 24 //numărul a este afișat pe 5 poziții c= 123.12 // numărul c este afișat pe 7 poziții cu 2 cifre după virgulă xxxxxxxx77</pre>
<i>Operații de intrare</i>	
Pascal	<pre>read(x); sau readln(x);</pre> <p><code>read(x1, x2, ..., xn)</code>; este echivalent cu <code>read(x1); read(x2,); ...read(xn);</code></p> <p>La execuția acestei instrucțiuni compilatorul așteaptă de la tastatură valoare (valori) pentru parametrul (parametrii) înscrisi în paranteze. La introducerea datelor trebuie să ținem cont de tipul parametrilor. În cazul a mai multor valori numerice ele se despart prin spațiu. Finalul introducerii este tasta enter.</p> <p>Se recomandă ca înainte de orice instrucțiune read să fie o instrucțiune write, unde se va scrie ce se dorește de introdus, excepție făcând cazul citirii din fișier. În caz că lista parametrilor este mai mare de 1 la introducerea lor pot fi despărțite și prin enter, în acest caz menționăm că se introduc în continuare, nu de la început. Pentru tipul de date string se recomandă să fie unicul parametru în listă sau măcar ultimul.</p> <p>Fie următoarea secvență de program:</p> <pre>Var a,b:integer;; Begin write ('Introduceti doua numere intregi ') readln(a,b);...</pre> <p>sau</p> <pre>Begin write ('Introduceti valoarea lui a ') readln(a);</pre>

	<code>write ('b=') readln(b); ...</code>
C	<p>Pentru introducerea datelor cu format se folosește funcția <i>scanf()</i>, este definită în biblioteca <i>stdio.h</i>. Mod de utilizare :</p> <pre>scanf(șir_de_control,lista_de_intrare);</pre> <p>unde :</p> <p><i>șir_de_control</i> este un șir scris în ghilimele. În cadrul acestui șir se indică tipul valorilor care urmează a fi citite. Șirul este format din % și caracterul de conversie (sunt date mai sus).</p> <p><i>lista_de_intrare</i> indică adresele variabilelor pentru care vor fi citite valori. În fața fiecărei variabile va fi operatorul adresă &. În cazul în care se citesc mai multe valori acestea vor fi despărțite prin virgulă.</p> <p><i>Exemplu</i></p> <p>Fie date următoarele declarații :</p> <pre>int a; float b;</pre> <p>Pentru citirea valorilor de la tastatură a acestor variabile într-o singură instrucțiune vom scrie :</p> <pre>scanf("%d%f", &a, &b);</pre> <p><i>%d</i> indică tipul variabilei a (<i>int</i>), pe când <i>%f</i> indică tipul variabilei b (<i>float</i>).</p>
C++	<p>Pentru citirea datelor de la consolă vom scrie :</p> <pre>cin>>var1>>var2>>...>>varn;</pre> <p>unde <i>var1</i>, <i>var2</i>, ..., <i>varn</i> sunt variabile.</p> <p><i>Exemplu</i></p> <p>Fie date următoarele declarații :</p> <pre>int a; float b;</pre> <p>Pentru citirea valorilor de la tastatură a acestor variabile într-o singură instrucțiune vom scrie :</p> <pre>cin>>a>>b;</pre>

Probleme pentru rezolvare independentă

1) De la tastatură se citește un caracter. Elaborați un program prin intermediul la ecran se va afișa un triunghi de forma

a)	b)	c)	d)
Date de intrare *	Date de intrare *	Date de intrare *	Date de intrare *
Date de ieșire	Date de ieșire	Date de ieșire	Date de ieșire
*	* * * *	****	*
**	* * *	***	**
***	* *	**	* * *
****	*	*	* * * *

2) Într-o tabără numărul de băieți este cu 10 mai mare decât cel al fetelor. De la tastatură se citește numărul de fete. Elaborați un program prin intermediul căruia se va determina numărul elevilor din tabără. Exemplu: date de intrare: 50 date de ieșire: 110.

3) Într-un autobuz care pleacă în excursie sunt 7 copii. De la încă două școli urcă alți copii, numărul acestora citindu-se de la tastatură. Elaborați un program prin intermediul căruia se va determina numărul copiilor care au plecat în excursie. Exemplu: Date de intrare: 15 20 Date de ieșire: 42 copii.

4) Un brăduț este împodobit cu globulețe albe, roșii și albastre. Numărul globulețelor albe se citește de la tastatură. Elaborați un program prin intermediul căruia se va determina numărul globulețelor din brăduțul, dacă numărul de globulețelor roșii este cu 3 mai mare decât numărul de globulețe albe, iar numărul globulețele albastre este mai mic cu 2 decât totalul celor albe și roșii. Exemplu: Date de intrare: 12 Date de ieșire: 52.

5) Ion și Vasile joacă după următoarele reguli: Ion spune un număr iar Vasile trebuie să găsească cinci numere consecutive, crescătoare, numărul din mijloc fiind cel ales de Ion. Elaborați un program prin intermediul căruia la ecran se vor afișa numerele găsite de către Vasile. Exemplu: Ion spune 10, Vasile spune 8 9 10 11 12.

6) Doi copii au primit același n mere, n se citește de la tastatură. Elaborați un program prin intermediul căruia la ecran se va afișa numărul de mere pe care îl au copii dacă primul copil mănâncă un măr și dă unul celuilalt copil. Exemplu: Date de intrare: 10 Date de ieșire: primul copil 8 mere al doilea copil 11 mere.

7) Maria vrea să verifice dacă greutatea și înălțimea ei corespund vârstei pe care o are. Ea a găsit într-o carte următoarele formule de calcul ale greutateii și înălțimii unui copil, v fiind vârsta: $\text{greutate}=2*v+8$ (în kg), $\text{înălțime}=5*v+80$ (în cm). Elaborați un program prin intermediul căruia se va determina greutatea și înălțimea ideală a unui copil, dacă vârsta se citește de la tastatură.

8) Se introduc de la tastatură trei cifre. Elaborați un program prin intermediul căruia se va afișa pe aceeași linie 5 numere formate cu aceste cifre luate o singură dată. Exemplu: date de intrare: 3 4 2 Date de ieșire: 324 342 243 234 432.

9) De la tastatură se citesc trei numere întregi. Elaborați un program prin intermediul căruia se vor determina toate sumele posibile de câte două numere. Afișarea va cuprinde și termenii sumei, nu numai valoarea ei. Exemplu: Date de intrare: 2 13 4 Date de ieșire: $2+13=15$ $2+4=6$ $13+4=17$.

10) De la tastatură se citește un număr întreg $n(1 \leq n \leq 10)$. Elaborați un program prin intermediul căruia se va afișa la ecran tabla înmulțirii cu numărul n . Exemplu: pentru $n=5$, se va afișa pe verticală $1 \times 5=5$ $2 \times 5=10$ $3 \times 5=15$ $4 \times 5=20$ $5 \times 5=25$ $6 \times 5=30$ $7 \times 5=35$ $8 \times 5=40$ $9 \times 5=45$ $10 \times 5=50$.

11) De la tastatură se citește un număr întreg, care reprezintă numărul de ani. Elaborați un program prin intermediul căruia la ecran se va afișa numărul de luni, zile și ore corespunzătoare. Se consideră că un an are 365 zile. Exemplu: date de intrare: 2 date de ieșire: 24 luni 730 zile 17520 ore.

12) De la tastatură se citesc două numere întregi a și b . Elaborați un program prin intermediul căruia să se schimbe între ele valorile variabilelor a și b . La ecran se vor afișa noile valori ale lui a și b .

13) * Rezolvați problema **12)** și fără a utiliza o variabilă intermediară.

14) Distanța dintre două orașe A și B este de x km. Un șofer cu automobilul, parcurge această distanță în a minute. Elaborați un program prin intermediul căruia la ecran se va afișa viteza medie a automobilului, dacă valorile lui x și a se citesc de la tastatură.

15) Distanța dintre două orașe A și B este de x km. Un șofer cu automobilul parcurge această distanță. Elaborați un program prin intermediul căruia se va determina costul deplasării, dacă la distanța de 100 km automobilul consumă în medie y litri de combustibil. Costul unui litru de combustibil, distanța dintre orașe și consumul la 100km, se citesc de la tastatură.

16) O navă parcurge distanța d (în km) dintre două stații orbitale în a săptămâni și b zile. Elaborați un program prin intermediul căruia se va determina viteza navei exprimată în km/oră.

§2 Funcții matematice

În varianta Pascal

Descriere	Funcția	Tipul argumentelor	Tipul rezultatului
Arctangenta valorii x	$arctan(x);$	Numeric, adică integer sau real	real
Sinusul lui x	$sin(x);$		
Cosinusul lui x	$cos(x);$		
Puterea numărului $e e^x$	$exp(x);$		
logaritmul natural lnx	$ln(x);$		
Rădăcina pătrată a lui x	$sqrt(x);$		
Valoarea absolută a lui x	$abs(x);$		
Pătratul lui x	$sqr(x);$		
Rotunjirea lui x	$round(x)$	real	integer
Trunchierea lui x	$trunc(x)$		
Paritatea lui x , True, dacă x este impar.	$odd(x)$	integer	boolean
Numărul valorii ordinale x	$ord(x)$	ordinal	integer
Predecesorul lui x	$pred(x)$	ordinal	Coincide cu tipul lui x
Succesorul lui x	$succ(x)$		
Caracterul cu numărul x	$chr(x)$	integer	char
Testarea sfârșitului de fișier	$eof(f)$	fișier	boolean
Testarea sfârșitului de linie	$eoln(f)$		
Partea fracționara obținuta prin trunchierea argumentului	$frac(x)$	real	real
Partea întreaga obținuta prin trunchierea argumentului	$Int(x)$	real	real
Valoarea numărului Pi (3,1415926536).	Pi	Numeric	real
Decrementarea valorii variabilei x cu 1 [respectiv n].	$Dec(x$ $[:n:LONGINT])$;	ordinal	Coincide cu tipul lui x
Incrementarea valorii variabilei x cu 1 [respectiv n].	$Inc(x:ordinal$ $[:n:LONGINT])$;	ordinal	Coincide cu tipul lui x
Inițializarea generatorului intern de numere aleatoare cu o valoare arbitrara.	$Randomize;$		
Număr real pseudoaleator, subunitar, nenegativ (fără argument).	$Random$	-	real
Număr întreg pseudoaleator din domeniul $0..Ord(x)-1$.	$Random(x)$	integer	WORD

În varianta C/C++

Antet	Descriere
<code>double asin(double arg);</code> <code>double acos(double arg);</code>	Calculează arcsinusul/arccosinusul valorii arg ; rezultatul este măsurat în radiani
<code>double atan(double arg);</code> <code>double atan2(double y, double x);</code>	Calculează arctangenta valorii arg , respectiv a fracției y/x
<code>double floor(double num);</code>	Întoarce cel mai mare întreg mai mic sau egal cu num (partea întregă inferioară)
<code>double ceil(double num);</code>	Întoarce cel mai mic întreg mai mare sau egal cu num (partea întregă superioară)
<code>double sin(double arg);</code> <code>double cos(double arg);</code> <code>double tan(double arg);</code>	Calculează sinusul/cosinusul/tangenta parametrului arg , considerată în radiani
<code>double sinh(double arg);</code> <code>double cosh(double arg);</code> <code>double tanh(double arg);</code>	Calculează sinusul/cosinusul/tangenta hiperbolică a parametrului arg
<code>double exp(double arg);</code>	Întoarce valoarea e^{arg}
<code>double pow(double base, double exp);</code>	Întoarce valoarea $base^{exp}$
<code>double log(double num);</code>	Calculează logaritmul natural (de bază e) al valorii arg
<code>double log10(double num);</code>	Calculează logaritmul în baza 10 al parametrului
<code>double sqrt(double num);</code>	Calculează rădăcina pătrată a parametrului
<code>double fmod(double x, double y);</code>	Întoarce restul împărțirii lui x la y
<code>double fabs(double arg);</code>	Întoarce valoarea absolută a lui arg

* *Argumentul sau argumentele obligatoriu se scriu în paranteze după numele funcției.*

Probleme pentru rezolvare independentă

1) Elaborați un program prin intermediul căruia se va determina valoarea expresiei:

I. $f(x) = x^3 + x^2;$

II. $f(x) = \frac{1}{x} + \frac{1}{x^2};$

III. $f(x) = \frac{|x|}{x^2 + 1};$

IV. $f(x) = x^2 \ln x;$

$$\text{V. } f(x) = \sin x + \frac{1}{2} \sin 2x ;$$

$$\text{VI. } f(x) = x^2 + \frac{8}{x} ;$$

$$\text{VII. } f(x) = e^{-x^2} ;$$

$$\text{VIII. } f(x) = \frac{e^x - e^{-x}}{2} ;$$

$$\text{IX. } f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} ;$$

$$\text{X. } f(x) = \frac{\operatorname{tg} x}{x} .$$

De la tastatură se citește valoarea lui x. Rezultatul va fi afișat cu o precizie de 3 cifre după virgulă.

2) De la tastatură se citesc valorile variabilelor x și y, care sunt numere reale. Elaborați un program prin intermediul căruia se va determina valoarea funcției f(x,y). Rezultatul va fi afișat la ecran cu o precizie de 4 cifre după virgulă:

$$\text{I. } f(x, y) = \frac{x^y + y^x}{x^{-y} + y^{-x}}$$

$$\text{II. } f(x, y) = \frac{\operatorname{tg}(x) + \operatorname{tg}(y)}{\sqrt{|x - y| + 1}}$$

$$\text{III. } f(x, y) = \frac{|x - y|}{x + \sin\left(\frac{\pi}{y}\right)} * \cos\left(\frac{\pi}{x}\right)$$

$$\text{IV. } f(x, y) = \sqrt{\frac{|x + y|}{\sqrt{\frac{1}{|x - y|}}}} * (|x| + 1)$$

$$\text{V. } f(x, y) = \frac{\sin(x) + \sin(y)}{e^y + e^x}$$

$$\text{VI. } f(x, y) = \sqrt[x]{y} + \sqrt[y]{x}$$

$$\text{VII. } f(x, y) = \frac{x - y}{|x + y| + 1} + \frac{x^2 - y^2}{x^2 + y^2 + 1}$$

$$\text{VIII. } f(x, y) = \frac{\cos^2(x) + 2 \sin^2(y)}{\pi}$$

$$\text{IX. } f(x, y) = \frac{x + y}{\sqrt{(x^2 + y^2 + 1) * \sin^2(x + y)}}$$

$$\text{X. } f(x, y) = \frac{\ln(x) + \ln(y)}{e^y + e^{-x}}$$

Instrucțiunea de ramificare if

If condiție then instrucțiunea1 else instrucțiunea2;

unde **condiție** este o expresie logică, care poate avea numai una din două valori *true* sau *false*. Dacă rezultatul este *true* se execută instrucțiunea 1 și se trece la următoarea instrucțiune după **if**, și dacă rezultatul este *false* se execută **instrucțiunea 2**. Menționăm aici că:

1. Dacă este necesar să fie îndeplinite mai mult de o instrucțiune se folosește instrucțiunea compusă *begin ...end*.
2. Înainte de cuvântul rezervat *else* nu se pune ;
3. Condiția poate fi simplă, dar și compusă folosind OR, AND, NOT. În acest caz trebuie să folosim parantezele.

Exemple:

```
If a>b then c:=a else c:=b;
c:=b; If a>b then c:=a; {Instrucțiunea de ramificare redusă}
```

```
If a>b
  then begin c:=a; writeln('a este mai mare');
        end
  else begin c:=b; writeln('b este mai mare');
        end;
```

```
If a>0
  then if a mod 2=0 then writeln('a este pozitiv, par')
        else writeln('a este pozitiv, impar')
  else if a mod 2=0 then writeln('a este negativ, par')
        else writeln('a este negativ, impar')
```

```
If (a>=10) and (a<100)
  then writeln('a este format din 2 cifre')
```

if...else

este cea mai simplă instrucțiune condițională.

Poate fi folosită în mai multe forme:

```
if( condiție ){ // instrucțiuni }
```

```
if( condiție ){ //instrucțiuni }
  else{ //alte instrucțiuni }
```

```
if( condiție1 ){//instrucțiuni1 }
  else if( condiție2 ){ //instrucțiuni2 }
    else if( condiție N ){ //instrucțiuni N }
```

Pascal

C/C++

Probleme pentru rezolvare independentă

- 1) Se introduc punctajele a doi sportivi. Elaborați un program prin intermediul căruia se va afișa punctajele în ordine descrescătoare. Exemplu: Date de intrare 100 134 Date de ieșire: 134 puncte 100 puncte
- 2) Se dau două numere. Elaborați un program prin intermediul căruia se va înmulți cel mai mare cu doi și cel mai mic cu trei și să se afișeze rezultatele. Exemplu: date de intrare: 3 7 date de ieșire: 9 14
- 3) Se introduc două numere întregi. Elaborați un program prin intermediul căruia se va determina dacă primul număr este predecesorul (succesorul) celui de-al doilea. Exemple: date de intrare: 2 4 date de ieșire: Nu; date de intrare: 5 6 date de ieșire: Da.
- 4) Ionel spune părinților doar notele mai mari sau egale cu 7. Într-o zi el a luat trei note. Elaborați un program prin intermediul căruia la ecran se vor afișa doar notele pe care le va comunica Ionel părinților. Exemplu : Date de intrare 8 7 5 Date de ieșire 8 7.
- 5) Se introduc două numere. Elaborați un program prin intermediul căruia se va afișa câtul dintre primul și al doilea, dacă al doilea număr este diferit de 0 sau mesajul “Împărțire imposibilă” în caz contrar. Exemple: Date de intrare 10 3 Date de ieșire 3.33 Date de intrare 45 0 date de ieșire Impartire imposibila.
- 6) Un ascensor pentru copii acceptă o greutate de maxim 100 kg. De la tastatură se introduc greutățile a doi copii. Elaborați un program prin intermediul căruia se va afișa mesajul POT INTRA AMBII COPII, dacă greutatea copiilor nu depășește 100 kg și mesajul INTRA PE RIND în caz contrar. Exemple: Date de intrare: greutăți copii 87 50 Date de ieșire : Intra pe rând Date de intrare 45 52 Date de ieșire Pot intra ambii copii.
- 7) Ionel are voie să se uite la TV 20 de ore pe săptămână fără a fi pedepsit. De la tastatură se citesc numărul de ore privite la TV pentru fiecare zi din săptămână. Elaborați un program prin intermediul căruia se va determina dacă Ionel va fi, sau nu pedepsit. Exemplu: Date de intrare: 3 4 2 2 5 6 1 Date de ieșire: Va fi pedepsit.
- 8) Elaborați un program prin intermediul căruia se va determina greutatea ideală a unei persoane cunoscând înălțimea, vârsta și sexul persoanei. Formulele de calcul sunt:

$$G_{\text{masculin}} = 50 + 0.75 * (\text{inaltime} - 150) + (\text{varsta} - 20) / 4,$$

$$G_{\text{feminin}} = G_{\text{masculin}} - 10,$$

unde înălțimea este exprimată în cm și vârsta în ani. Sexul se citește sub forma unui caracter, f sau m. Exemplu: Date de intrare: inaltime= 160 varsta=21 sex=f Date de ieșire: greutate= 47.75 kg.

9) De la tastatură se citesc trei numere întregi. Elaborați un program prin intermediul căruia în dreapta fiecărui număr va fi afișat unul dintre mesajele: PAR, dacă numărul este par și IMPAR în caz contrar. Exemplu : Date de intrare: 45 3 24 Date de ieșire: 45 impar 3 impar 24 par.

10) Într-o tabără, băieții sunt cazați câte 4 într-o căsuță, în ordinea sosirii. Ionel a sosit al n-lea, n se citește de la tastatură. Elaborați un program prin intermediul căruia se va determina în a câta căsuță se va afla Ionel Exemplu : date de intrare : n=69 date de ieșire : casuta 17.

11) Elevii clasei a V-a se repartizează în clase câte 25 în ordinea mediilor clasei a IV-a. Radu este pe locul x ($1 \leq X \leq 125$) în ordinea mediilor, x se citește de la tastatură. Elaborați un program prin intermediul căruia se va determina în ce clasa va fi repartizat (A, B, C, D sau E)?. Exemplu : date de intrare : x=73 date de ieșire : C.

12) De la tastatură se citesc trei numere diferite. Elaborați un program prin intermediul căruia:

i. se va afișa cel mai mare și cel mai mic număr. Exemplu : Date de intrare 45 34 78 Date de ieșire max=78 min=34. Numerele vor fi afișate în ordine crescătoare. Exemplu : Date de intrare 4 2 6 Date de ieșire 2 4 6.

ii. se va afișa numărul a cărei valoare este cuprinsă între valorile celorlalte două. Exemplu: Date de intrare 12 14 10 Date de ieșire 12.

iii. se va verifica dacă numerele introduse, formează o secvență de numere consecutive. Exemple: Date de intrare 3 4 5 Date de ieșire Da Date de intrare 4 5 7 Date de ieșire Nu.

13) Andrei primește într-o zi trei note, nu toate bune. Se hotărăște ca, dacă ultima notă este cel puțin 8, să le spună părinților toate notele primite iar dacă este mai mică decât 8, să le comunice doar cea mai mare notă dintre primele două. De la tastatură se citesc notele primite de către Andrei. Elaborați un program prin intermediul căruia se vor afișa notele pe care Andrei le va comunica părinților. Exemple : Date de intrare 6 9 9 Date de ieșire 6 9 9 ; Date de intrare 8 5 7 Date de ieșire 8.

14) De la tastatură se citesc trei numere. Elaborați un program prin intermediul căruia se va afișa numărul mai mare dintre al doilea și al treilea număr, dacă toate trei numere sunt pare și suma primelor două numere în caz contrar. Exemple: Date de intrare 46 32 100 date de ieșire 100 ; Date de intrare 34 -25 10 Date de ieșire 9.

15) Să se afișeze cel mai mare număr par dintre doua numere introduse în calculator. Exemple : Date de intrare 23 45 Date de ieșire nu exista numar par ; Date de intrare 28 14 Date de ieșire 28 ; Date de intrare 77 4 Date de ieșire 4.

16) De la tastatură se citesc trei cifre diferite de 0. Elaborați un program prin intermediul căruia se va afișa numărul format din aceste cifre, astfel încât acesta să fie maximal. Exemplu: Date de intrare : 4 5 3 Date de ieșire: 543.

17) De la tastatură se citesc trei cifre. Elaborați un program prin intermediul căruia se va afișa numărul format din aceste cifre, astfel încât acesta să fie minimal. Exemplu: Date de intrare : 4 5 3 Date de ieșire: 345. Notă: Participă și cifra 0, dar numărul nu poate începe cu 0.

§4 Instrucțiunea cu selecție multiplă

<i>Instrucțiunea cu selecție multiplă</i>	
Pascal	<pre> Case a of cazul 1 :instrucțiunea 1; cazul 2 :instrucțiunea 2; ... cazul n :instrucțiunea n [else instrucțiunea n+1;] end;</pre> <p>Menționăm, că:</p> <ol style="list-style-type: none"> 1. else poate lipsi; 2. Cazul poate fi o valoare, mai multe – despărțite prin virgulă, un subdomeniu cum și o combinație a acestora; 3. Variabila a trebuie să fie de tip ordinal; 4. Dacă <i>a</i> se regăsește în unul din cazuri – se execută instrucțiunea respectivă și se trece la următoarea instrucțiune, după end.
C/C++	<p>switch este o instrucțiune menită să simplifice structurile condiționale cu mai multe condiții.</p> <pre> switch(expresie){ case constanta1: //instrucțiuni1 break; case constanta2: //instrucțiuni2 break; [default: //instrucțiuni] }</pre> <p>Valoarea <i>expresie</i> de tip ordinal este evaluată la un tip întreg, apoi această valoare este comparată cu fiecare constantă; este rulat blocul de instrucțiuni al valorii găsite. În caz ca numărul nu este egal cu nici una dintre constante, este executat blocul aflat după default. În cazul, când lipsește break se verifică și celelalte cazuri.</p>

Probleme pentru rezolvare independentă

1) De la tastatură se citește un caracter (literă). Elaborați un program care va afișa textul E VOCALA, dacă caracterul introdus este vocală. De exemplu: date de intrare E, date de ieșire E VOCALA și NU E VOCALA în caz contrar.

2) De la tastatură se citește o cifră. Elaborați un program care va afișa cifra scrisă cu caractere. De exemplu: date de intrare 5, date de ieșire cinci

3) De la tastatură se citesc 2 numere întregi și un operator (+ - * / %). Elaborați un program care în dependență de datele introduse va afișa rezultatul. Dacă se va introduce un operator necunoscut rezultatul va fi 0. De exemplu: date de intrare 8 3 %, $8\%3=2$

4) De la tastatură se citesc 2 numere reale și un operator (+ - * /). Elaborați un program care în dependență de datele introduse va afișa rezultatul. Dacă se va introduce un operator necunoscut rezultatul va fi 0. De exemplu: date de intrare 8 3 *, $8*3=24$

5) De la tastatură se citește un număr pozitiv, mai mic sau egal cu 100. Elaborați un program care va afișa numărul citit cu cifre romane. De exemplu: date de intrare 18, date de ieșire XVIII

6) Magazinul comercial Nr 1, oferă clienților fideli o gamă largă de reduceri la produsele procurate. Dacă suma este mai mică decât 500 lei, atunci în zilele de odihnă reducerea este de 5%, iar în zilele de lucru 3,5%. Dacă suma este mai mare sau egală cu 500 lei atunci reducerile se oferă respectiv: 4% Luni, Marți 5%, Miercuri 6%, Joi 7%, Vineri, 8% Sâmbătă 9%, Duminică 10%. Elaborați un program care în dependență de ziua și suma introdusă va afișa la ecran reducerea oferită, cât și Suma ce necesită a fi achitată. De exemplu: date de intrare Duminică 1000, date de ieșire reducere 100 lei, suma ce necesită a fi achitată 900 lei.

§5 Instrucțiuni ciclice

În varianta Pascal	În varianta C/C++
<p>while execută un bloc de instrucțiuni atât timp cât o anumită condiție este adevărată. Forma generală a unui ciclu while este:</p>	
<pre>while (expresie) do instrucțiune; while (expresie) do begin instrucțiuni end;</pre>	<pre>while (expresie){ //instrucțiuni }</pre>
<p>Dacă condiția din <i>expresie</i> este adevărată (are o valoare nenulă), instrucțiunile din blocul de după <i>while</i> sunt executate. <i>expresie</i> este reevaluată după fiecare iterație. Pentru ca condiția să devină falsă este în responsabilitatea programatorului.</p>	
<i>Repeat</i> <i>until</i> (expresie);	do ... while (expresie);
<p>este o instrucțiune repetitivă similară cu cea precedentă, singura diferență fiind că <i>expresie</i> este evaluată după executarea instrucțiunilor, nu înainte. Astfel, blocul va fi executat cel puțin o dată.</p>	
<pre><i>Repeat</i> <i>Instrucțiuni</i> <i>.until</i> (expresie);</pre>	<pre>do{ //instrucțiuni } while (expresie);</pre>
<p>for reprezintă o formă mai simplă de a scrie un while însoțit de o expresie inițială și una finală. Deosebim 2 variante: una cu pasul +1: <i>for i:=1 to 10 do</i> instrucțiune; alta cu pasul -1 <i>for i:=10 downto 1 do</i> instrucțiune;</p>	<p>for reprezintă o formă mai simplă de a scrie un while însoțit de o expresie inițială și de o expresie de incrementare. Forma sa este: for(expresie1;expresie2;expresie3){ //instrucțiuni }</p>
<p>Exemplul următor prezintă un ciclu cu funcționalitate identică (tipărirea primelor 10 numere naturale), folosind cele 3 instrucțiuni repetitive:</p>	
<pre>Var i:integer; Begin Writeln('Ciclul for '); For i:=1 to 10 do write (i:3); Writeln; Writeln('Ciclul while '); i:=1; while i<=10 do begin write (i:3); i:=i+1; end; Writeln; Writeln('Ciclul repeat ..until'); i:=1; repeat write (i:3); i:=i+1; until i>10;</pre>	<pre>int main(){ int i; printf("Ciclul for\n"); for(i=1;i<=10;i++) printf("i=%d\n", i); printf("Ciclul while\n"); i=1; while (i <= 10) { printf("i=%d\n", i); i++; } printf("Ciclul do while\n"); i=0; do{ i++; printf("i=%d\n", i); }while(i < 10); }</pre>

end.

Probleme pentru rezolvare independentă

1) Elaborați un program prin intermediul căruia se vor afișa toate numerele de forma $a23a$ care se împart exact la 6.

2) De la tastatură se citesc două numere pozitive a și n . Elaborați un program prin intermediul căruia se va afișa numărul a urmat de n zerouri. Exemplu: Date de intrare $a=34$ $n=5$ Date de ieșire 3400000.

3) De la tastatură se citesc numărul n , $n(1 \leq n \leq 10)$. Elaborați un program prin intermediul căruia se va afișa Să se afișeze tabla înmulțirii cu n . Exemplu: Date de intrare $n=5$ date de ieșire $1 \times 5=5$ $2 \times 5=10$ $3 \times 5=15$ $4 \times 5=20$ $5 \times 5=25$ $6 \times 5=30$ $7 \times 5=35$ $8 \times 5=40$ $9 \times 5=45$ $10 \times 5=50$.

4) De la tastatură se citește un număr întreg pozitiv. Elaborați un program prin intermediul căruia se vor afișa toți divizorii unui număr natural citit. Exemplu: Date de intrare 12 Date de ieșire 1 2 3 4 6 12.

5) De la tastatură se citește un număr întreg pozitiv. Elaborați un program prin intermediul căruia se va verifica dacă este numărul este prim. Exemple: Date de intrare 23 date de ieșire Este Prim Date de intrare 45 Date de ieșire Nu este prim.

6) Un ascensor parcurge distanța dintre două etaje a și b . Elaborați un program prin intermediul căruia se vor afișa toate etajele parcurse, în ordinea atingerii lor. Exemple: Date de intrare $a=4$ $b=7$ Date de ieșire 4 5 6 7; Date de intrare $a=10$ $b=8$ Date de ieșire 10 9 8.

7) De la tastatură se citesc n numere întregi pozitive, fiind date numerele a , b și c . Elaborați un program care să afișeze toate numerele care se divid cu a sau b și sunt mai mici decât c .

8) De la tastatură se citește numărul n , întreg pozitiv. Elaborați un program prin intermediul căruia se vor determina sumele:

$$s1=1+2+\dots+n;$$

$$s2=1+3+\dots+2n-1;$$

$$s3=2+4+\dots+2n;$$

$$s4=1*2+2*3+3*4+\dots+(n-1)*n;$$

$$s5=1+1*2+1*2*3+\dots+1*2*3*\dots*n;$$

$$s6=2-3+4-5+\dots-99+100;$$

9) Se citesc pe rând temperaturile medii ale fiecărei luni a unui an, ca numere întregi. Elaborați un program prin intermediul căruia se va afișa cu două zecimale media anuală a temperaturilor pozitive și a celor negative. Exemplu: Date de intrare -5 -3 1 8 12 17 20 21 18 10 6 -2 Date de ieșire medie_poz=13.66 medie_neg=-3.33.

10) Se citesc mediile a n elevi, ca numere reale. Elaborați un program prin intermediul căruia se va afișa cea mai mare și cea mai mică medie. Să se verifice dacă sunt corigenți. Exemplu : Date de intrare $n=4$ 9.50 4.25 9.66 6.33 Date de ieșire max=9.66 min=4.25 1 corigent.

11) Se dau două numere întregi nenule. Elaborați un program prin intermediul căruia se va afișa cmmdc și cmmmc al lor. Exemplu : Date de intrare 12 32 Date de ieșire cmmdc=4 cmmmc 96.

12) Se dau trei numere. Elaborați un program prin intermediul căruia se va afișa cmmmdc al lor. Exemplu : Date de intrare 12 32 36 Date de ieșire 4.

13) Se citesc de la tastatură patru numere întregi diferite de zero. Numerele reprezintă în ordinea citirii: numărătorul și numitorul primei fracții, respectiv numărătorul și numitorul celei de a doua fracții. Elaborați un program prin intermediul căruia se va afișa suma celor două fracții. Exemplu: Date de intrare: 6 18 12 24 Date de ieșire: 5 6.

14) De la tastatură se citesc 2 numere nenegative mai mici decât 10, care reprezintă lungimile laturilor unui dreptunghi. Elaborați un program care în dependență de datele introduse va afișa un dreptunghi. Exemplu:

a)	b)
Date de intrare 4 3	Date de intrare 4 3
Date de ieşire	Date de ieşire
1111	1234
2222	1234
3333	1234

15) Se introduc succesiv numere nenule întregi până la introducerea numărului 0. Să se afișeze suma tuturor numerelor pare introduse. Exemplu: Date de intrare 3 5 4 2 0 Date de ieşire 6.

16) Se introduc succesiv numere nenule întregi până la introducerea numărului 0. Să se afișeze suma tuturor numerelor divizibile cu 3 introduse. Exemplu: Date de intrare 3 5 6 2 0 Date de ieşire 9.

17) De la tastatură se citește un număr întreg nenegativ n , mai mic decât 10. Elaborați un program care va afișa un triunghi.

a)	b)
Date de intrare 4	Date de intrare 4
Date de ieşire	Date de ieşire
1	1234
12	123
123	12
1234	1

18) Se citesc numere de la tastatură până la introducerea unui număr impar divizibil cu 3. Să se afișeze suma tuturor numerelor pare și a numerelor impare introduse. Precizați câte numere au fost introduse. Exemplu: Date de intrare 7 4 5 2 1 9 Date de ieşire suma pare =6 suma impara=23.

19) Se citesc numere de la tastatură până la introducerea unui număr divizibil cu 5. Să se afișeze suma tuturor numerelor pare și a numerelor impare introduse. Precizați câte numere au fost introduse. Exemplu: Date de intrare 7 4 6 2 1 25 Date de ieşire suma pare =12 suma impara=35.

20) Pentru a o elibera pe Ileana Cosânzeana, Făt-frumos trebuie să parcurgă x km. El merge zilnic a km, dar Zâna-cea-Rea îl duce în fiecare noapte cu b km înapoi, $b < a$. Elaborați un program prin intermediul căruia se va afișa după câte zile Făt-frumos

o eliberează pe Ileana Cosânzeana. Exemplu: Date de intrare $x=10$ $a=4$ $b=1$ Date de ieșire 3 zile.

21) Se citesc de la tastatură numere naturale până când suma numerelor pare este mai mare decât k . Câte numere au fost introduse și care este suma numerelor pare? Exemplu: Date de intrare: $K=12$ 5 8 1 2 2 3 6 Date de ieșire: 7 (numere) 18 (suma celor pare).

22) Se citesc de la tastatură numere naturale până când suma numerelor pare este mai mare decât k . Câte numere au fost introduse și care este suma numerelor impare? Exemplu: Date de intrare: $K=12$ 5 8 1 2 2 3 6 Date de ieșire: 7 (numere) 9 (suma celor impare)

23) Se citesc de la tastatură numere întregi pozitive atât timp cât suma lor nu depășește 1000. Să se scrie un program care să afișeze cea mai mică și cea mai mare valoare a acestor numere. Exemplu: 550 345 100 45 Date de ieșire $\max=550$ $\min=100$.

24) Se citesc de la tastatură numere întregi pozitive atât timp cât suma lor nu este divizibilă cu 3 Să se scrie un program care să afișeze cea mai mică și cea mai mare valoare a acestor numere. Exemplu: 7 3 6 2 Date de ieșire $\max=7$ $\min=2$.

25) De la tastatură se citesc n numere întregi pozitive. Elaborați un program care va determina suma și media numerelor citite cu excepția numerelor unde suma cifrelor este un număr impar.

26) De la tastatură se citesc n numere întregi pozitive. Elaborați un program care va determina suma și media numerelor citite cu excepția numerelor unde suma cifrelor cărora este un număr divizibil cu trei.

27) De la tastatură se citesc datele despre n ($n>2$) dreptunghiuri (lungimea și lățimea). Elaborați un program care va afișa dreptunghiul (lungimile laturilor, aria și perimetru) cu suprafața maximă și dreptunghiul cu cel mai mic perimetru.

28) De la tastatură se citesc datele despre n ($n>2$) triunghiuri dreptunghice (lungimea catetelor). Elaborați un program care va afișa triunghiul (lungimile laturilor aria și perimetru) cu suprafața maximă și triunghiul cu cea mai mică ipotenuză.

29) Elaborați un program care va afișa toate numerele întregi pozitive a, b, c , mai mici ca 20, cu următoarea proprietate: $a^2+b^2=c^2$

30) Elaborați un program care va afișa toate numerele întregi pozitive a, b, c , mai mici ca 20, cu următoarea proprietate: $(a+b)^3=c^3$

31) De la tastatură se citesc trei numere întregi pozitive a, b, c mai mici ca 100000. Să se afișeze toți divizorii comuni.

32) De la tastatură se citesc trei numere întregi pozitive a, b, c mai mici ca 10000. Să se afișeze la ecran:

- a. numărul care are suma cifrelor o valoare maximă;
- b. numărul care are produsul cifrelor o valoare maximă;
- c. multiplu comun ale acestor numere.

§6 Recapitulare

1) Se consideră numărul întreg N . Scrieți un program care determină numărul de cifre impare și numărul de cifre pare ale acestui număr (zero considerându-se număr par).

2) Se consideră N un număr întreg. Scrieți un program care determină dacă în scrierea zecimală a numărului N cifrele formează o consecutivitate strict crescătoare.

De exemplu: pentru N la Intrare: **257** la ieșire va fi afișat cuvântul **Da**
și dacă N la Intrare este **275** la ieșire va fi afișat cuvântul **Nu**

3) Scrieți un program care să determine toate numerele naturale mai mici decât **MAXINT** cu proprietatea că sunt egale cu suma factorialilor cifrelor lor.

Exemplu: $145 = 1! + 4! + 5!$

4) Scrieți un program care determină toate numerele naturale mai mari ca 1 și mai mici decât un număr natural diferit de zero dat și prime cu el numărând câte sunt. (Notă: două numere naturale se numesc prime între ele dacă cel mai mare divizor comun ale acestora este 1) Valoarea numărului respectiv este citită de la tastatură, răspunsul se afișează la ecran, pe primul loc – numărul ce indică cantitatea de numere cu proprietatea dată, și în continuare – numerele găsite.

De exemplu dacă este citit numărul 20 se va afișa 7 3 7 9 11 13 17 19

5) Un număr de trei cifre se numește „*norocos*” dacă suma cărorva două cifre ale acestui număr este egală cu a treia. Elaborați un program care citește de la tastatură n numere, formate din trei cifre și determină numărul de numere „*norocoase*”

6) Un număr de patru cifre se numește „*norocos*” dacă suma primelor două cifre este egală cu suma ultimelor două. Elaborați un program care citește de la tastatură n numere, formate din patru cifre și determină numărul de numere „*norocoase*”

7) Un număr de patru cifre se numește „*nenorocos*” dacă suma primelor două cifre este egală cu suma ultimelor două și este 13. Elaborați un program care citește de la tastatură n numere, formate din patru cifre și determină numărul de numere „*nenorocoase*”

8) Scrieți un program care citește de la tastatură numărul natural N și determină numărul minim care poate fi obținut din toate cifrele acestui număr. De exemplu, dacă $N=24175$, atunci rezultatul va fi.: 12457. Scrierea cifrelor nu se începe cu cifra zero.

Capitolul 2

TIPURI STRUCTURATE

§1 Tablouri unidimensionale

1) De la tastatură se citesc n ($4 < n < 1000$) numere întregi. Elaborați un program prin intermediul căruia se vor efectua următoarele operații:

- a) Se vor afișa la ecran numerele la un interval de 5 poziții;
- b) Se vor afișa la ecran numerele în ordinea inversă a introducerii lor la un interval de 4 poziții;
- a) Se vor afișa doar numerele pare;
- b) Se vor afișa doar numerele impare;
- c) Se vor afișa doar numerele mai mari ca x și nu sunt divizibile cu y , x și y se citesc de la tastatură;
- d) Se vor afișa doar numerele mai mari ca x și mai mici decât y , x și y se citesc de la tastatură;
- e) Se vor afișa pozițiile numerelor negative și impare din tablou;
- f) Se va afișa media numerelor pare;
- g) Se vor afișa pozițiile numerelor formate din 2 cifre;
- h) Se va afișa suma numerelor divizibile cu 3;
- i) Se va afișa cele mai mari 2 numere;
- j) Se va afișa toate numerele cu excepția celui minimal și maximal.

2) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program care va afișa elementele introduse, apoi va mari fiecare element al tabloului cu 10. Elementele modificate se vor afișa la ecran.

3) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program prin intermediul căruia elementele pare ale vectorului, se vor dubla, iar cele impare se vor tripla. Se vor afișa elementele vectorului modificat.

4) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program prin intermediul căruia primul element din tablou va fi înlocuit cu elementul minimal și invers. Se vor afișa elementele vectorului modificat.

5) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program prin intermediul căruia al treilea element va fi înlocuit cu elementul maximal și invers. Se vor afișa elementele vectorului modificat.

6) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program prin intermediul căruia se va crea un nou vector care va fi format numai din elementele vectorului inițial care au exact cel puțin trei divizori. Se vor afișa elementele ambilor vectori.

7) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program care va afișa elementele introduse, apoi va micșora fiecare element al tabloului cu 12. Elementele modificate se vor afișa la ecran.

8) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program prin intermediul căruia:

i. elementele negative ale vectorului, se vor dubla, iar cele mai mari decât 10 se vor micșora cu 15. Se vor afișa elementele vectorului modificat.

ii. se va crea un nou vector care va fi format numai din elementele impare ale vectorului inițial. Se vor afișa elementele ambelor vectori.

iii. care va afișa elementele introduse, apoi se va mari fiecare element al tabloului de 5 ori. Elementele modificate se vor afișa la ecran.

iv. elementele impare ale vectorului, se vor mări cu 25, iar cele negative micșora de 2 ori. Se vor afișa elementele vectorului modificat.

v. elementul maximal al vectorului, va fi înlocuit cu cel minimal și reciproc. Se vor afișa elementele vectorului modificat.

9) De la tastatură se citesc elementele unui vector de dimensiune n ($n < 1000$) cu numere întregi. Elaborați un program prin intermediul căruia se va crea un nou vector care

i. va fi format numai din elementele pare ale vectorului inițial. Se vor afișa elementele ambilor vectori.

ii. va fi format numai din elementele vectorului inițial care au doi sau patru divizori. Se vor afișa elementele ambilor vectori.

iii. va fi format numai din elementele divizibile cu 3 ale vectorului inițial. Se vor afișa elementele ambilor vectori.

iv. va fi format numai din elementele vectorului inițial care au cel mult patru divizori. Se vor afișa elementele ambilor vectori.

§2 Tablouri bidimensionale

1) De la tastatură se citesc elementele unui tablou bidimensional cu n – linii și m – coloane $n, m < 10$ de numere întregi. Elaborați un programa care:

- a) Va afișa elementele matricei;
- b) Va determina numărul elementelor pare, mai mici decât 10;
- c) Va determina numărul elementelor formate dintr-o singură cifră;
- d) Va înlocui elementele divizibile cu 3, cu valoarea numărului maximal *3, și va afișa matricea modificată;
- e) Va înlocui elementele pare cu valoarea numărului maximal * 2, și va afișa matricea modificată;
- f) Va afișa elementele de pe coloana 2 și elementele de pe linia 1;
- g) Va afișa elementele de pe coloana 1 și elementele de pe linia 2;
- h) Va afișa elementele de pe coloana a și elementele de pe linia b.

2) Să se construiască un tablou pătratic de dimensiune n, n cu primele $n*n$ numere:

a. pare. Exemplu: $n=3$ se va afișa

2 4 6

8 10 12

14 16 18

b. impare. Exemplu: $n=3$ se va afișa

1 3 5

7 9 11

13 15 19

c. divizibile cu 5. Exemplu: $n=3$ se va afișa

5 10 15

20 25 30

35 40 45

3) De la tastatură se citesc elementele unui tablou bidimensional cu n linii și m coloane. Elaborați un program care va afișa la ecran elementele matricei, va adăuga o linie nouă, elementele cărora vor fi:

a. maximul de pe fiecare coloană. Rezultatul se va afișa la ecran.

Exemplu Pentru o matrice cu 3 lini și 3 coloane

1 9 3

8 5 6

7 8 19

Linia nouă adăugată va fi 8 9 19

- b. suma elementelor de pe coloana respectivă. Rezultatul se va afișa la ecran.
Exemplu Pentru o matrice cu 3 linii și 3 coloane
1 2 3
4 5 6
7 8 9
Linia nouă adăugată va fi 12 15 18
- c. suma dintre elementele primei linii și celei de a doua linie. Rezultatul se va afișa la ecran. Exemplu Pentru o matrice cu 2 linii și 3 coloane
1 2 3
4 5 6
Linia nouă adăugată va fi 5 7 9, adică $5=1+4$; $7=2+5$; $9=3+6$.
- 4) De la tastatură se citesc elementele unui tablou bidimensional cu n linii și n coloane. Elaborați un program care va afișa la ecran elementele tabloului și:
- A) va mări elementele de pe diagonala principală cu 5. Rezultatul se va afișa la ecran;
 - B) va afișa elementele de pe linia ce conține elementul minimal;
 - C) va mări elementele unei coloane cu 15. Numărul de ordine a coloanei se citește de la tastatură. Rezultatul se va afișa la ecran.
- 5) De la tastatură se citesc elementele unui tablou bidimensional cu n linii și m coloane. Elaborați un program care va afișa la ecran elementele tabloului și va mări elementele unei linii cu 10. Numărul de ordine a liniei se citește de la tastatură. Rezultatul se va afișa la ecran.
- 6) De la tastatură se citesc elementele unui tablou bidimensional cu n linii și m coloane. Elaborați un program care va afișa la ecran elementele tabloului și va afișa elementele de pe linia suma elementelor cărora este maximă.
- 7) De la tastatură se citesc elementele unui tablou bidimensional cu n linii și m coloane. Elaborați un program care va afișa la ecran elementele tabloului și va afișa elementele de pe coloana ce conține elementul maximal.

§3 Șiruri de caractere

Funcții și proceduri de prelucrare a șirurilor de caractere

În varianta Pascal

Antet	Descriere
writeln(s);	extrage la ecran șirului s
readln(s);	citește șirul de la tastatură în variabila s
LENGTH(S)	Lungimea șirului S
CONCAT(s1[,s2,...,sn])	Concatenează șirurile S1, S2, ...Sn. Echivalentă cu S1+S2+...+Sn.
POS(Subșir,Șir)	Obținem poziția apariției subșirului în șir. Dacă rezultatul e 0 rezultă că așa subșir nu există în șir.
COPY (Șirul, De_La_Care_Poziție, Câte_Carectere)	Obținem un subșir, din șirul dat, începând de la care poziție, de lungimea Câte_Carectere.
INSERT(Ce_Inseram, In_Care_Șir, De_la_care_Poziție)	Se inserează un subșir într-un șir, începând cu poziția indicată.
DELETE (Șirul, De_La_Care_Poziție, Câte_Carectere)	Din șirul dat se exclud de la poziția indicată numărul indicat de caractere.
VAL(Șir, Variabilă_Numerică, Cod)	Transformă șirul de caractere în număr. Dacă nu este posibil – cod este diferit de zero, și dacă s-a transformat cu succes – Cod este 0.
STR(Număr[:m[:n]]; Șir)	Transformă numărul în șir de caractere. M și n precizează formatul în șir.
UpCase(x);	Litera mare corespunzătoare literei mici date ca argument

Exemple:

Funcția sau procedura	Rezultatul
POS('ma', 'Informatica')	6
POS('in', 'Informatica')	0
COPY ('Informatica', 7, 3)	ati
INSERT ('m', 'maa', 3)-	mama
DELETE ('Informatica', 5, 4)	Infoica

În varianta C/C++

Antet	Descriere
puts(sir);	extrage la consolă conținutul șirului sir
gets(sir);	citește caractere de la tastatură în variabila sir
int strcmp(char *s1, char *s2);	compară șirurile s1 și s2 returnează o valoare pozitivă, dacă șirul s1>s2, 0 dacă s1 este egal cu s2 și o valoare negativă dacă s1<s2

<code>int strcmpi(char *s1, char *s2);</code>	similară cu <code>strcmp</code> majuscule sunt ne semnificative
<code>int strncmp(char *s1, char *s2, int n);</code>	similară cu <code>strcmp</code> , dar se compară șirurile <code>s1</code> și <code>s2</code> pentru cel mult <code>n</code> caractere din fiecare șir
<code>char *strcpy(char *d, char *s);</code>	copiază șirul sursă <code>s</code> în șirul destinație <code>d</code> și returnează adresa șirului destinație
<code>char *strncpy(char *d, char *s, int n);</code>	copiază maxim <code>n</code> caractere de la sursă <code>s</code> la destinația <code>d</code> și returnează adresa șirului destinație
<code>int strlen(char *s);</code>	returnează lungimea șirului fără a număra caracterul terminator <code>'\0'</code>
<code>char *strcat(char *d, char *s);</code>	concatenează cele două șiruri și returnează adresa șirului rezultat
<code>double atof(sir);</code>	convertește un șir către tipul double . Dacă această conversie eșuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0
<code>long double _atold(sir);</code>	convertește un șir către tipul long double . Dacă această conversie eșuează, valoarea întoarsă este 0
<code>int atoi(sir);</code>	convertește un șir către tipul int . Dacă această conversie eșuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0
<code>long atol(sir);</code>	convertește un șir către tipul long . Dacă aceasta conversie eșuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0.
<code>itoa(int valoare, sir, int baza);</code>	convertește o valoare de tip int în șir, care este memorat în variabila <code>sir</code> . baza reține baza de numerație către care să se facă conversia
<code>ltoa(long valoare, sir, int baza);</code>	convertește o valoare de tip long int în șir, care este memorat în variabila <code>sir</code>
<code>ultoa(unsigned long valoare, sir, int baza);</code>	convertește o valoare de tip unsigned long în șir, care este memorat în variabila <code>sir</code>

Probleme pentru rezolvare independentă

1) De la tastatură se citește un șir de caractere. Elaborați un program care va determina:

- A) numărul de majuscule din șir;
- B) numărul de cifre din șir;
- C) numărul de caractere speciale (paranteze, operatori aritmetici) din șir;
- D) numărul de cifre pare citite.

2) De la tastatură se citește un șir de caractere. Elaborați un program care va înlocui:

- A) cifrele din șir cu semnul +;
- B) vocalele din șir cu spațiu;
- C) majusculele din șir cu cifra 8.

3) De la tastatură se citește un șir de caractere. Elaborați un program care va afișa:

- i. doar cifrele din șir;
- ii. fiecare cuvânt din rând nou;
- iii. doar consoanele din șir;
- iv. toate cifrele cu excepția celor divizibile cu 3.

4) De la tastatură se citesc mai multe cifre. Elaborați un program prin intermediul căruia se va determina dacă măcar un număr care poate fi format din cifrele citite este par.

5) De la tastatură se citesc patru cuvinte, fiecare cuvânt fiind citit într-o singură variabilă. Elaborați un program care va forma o frază, va include toate cuvintele în șir. Fiecare cuvânt va fi despărțit prin spațiu, ultimul caracter din frază va fi semnul punct. La ecran se vor afișa cuvintele citite cât și fraza formată.

6) De la tastatură se citesc patru cuvinte, fiecare cuvânt fiind citit într-o singură variabilă, un cuvânt va fi format din minim 3 caractere. Elaborați un program care va forma un cuvânt nou, în felul următor: din primul cuvânt va adăuga primele 2 caractere, din al doilea cuvânt va adăuga primul caracter, primele trei caractere din cuvântul al treilea și $n/2$ caractere din cuvântul patru (n – lungimea cuvântului). La ecran se vor afișa cuvintele citite cât și cuvântul format.

7) De la tastatură se citesc patru cuvinte, fiecare cuvânt fiind citit într-o singură variabilă. Elaborați un program care va afișa cuvintele în ordine alfabetică. Fiecare cuvânt va fi afișat din rând nou.

§4 Fișiere

Pascal	Deosebim două feluri de fișiere: textuale (cu acces secvențial) și tipizate sau binare (cu acces direct). Instrucțiunile de lucru cu fișierele:		
	Instrucțiunea	textuale	binare
	Declararea în secțiunea <i>var</i>	F:text	F:integer sau altul oricare tip în afară de fișier
	Asignarea fișierului logic, declarat în program cu cel fizic, salvat pe disc. De obicei fișierele se află în dosarul activ, în caz contrar – se scrie calea completă a fișierului.	Assign (f, 'date.txt'); Assign (f, 'c:\valori\date.txt');	
	Deschiderea fișierului pentru citire și modificări	Reset(f);	
	Deschiderea unui fișier nou. Dacă așa fișier există – informația din el va dispărea.	Rewrite(f);	
	Citirea valorilor a și b din fișierul f	Read(f, a, b)	
	Citirea valorilor a și b din fișierul f și trecerea la linia următoare	Readln(f, a, b)	Nu există noțiunea de linie!!!
	Înscrierea valorilor a și b în fișierul f	write(f, a, b)	
	Înscrierea valorilor a și b în fișierul f și trecerea la linia următoare	writeln(f, a, b)	Nu există noțiunea de linie!!!
	Închiderea fișierelor. (Obligatorie în cazul când înscriu sau se modifică date).	Close(f)	
	<p>Citirea datelor din fișierul textual poate fi organizată în mai multe moduri, în dependență de condițiile problemei.</p> <p><i>Exemplu 1:</i> Fișierul <i>tablou.txt</i> conține în prima linie 2 numere întregi – <i>m</i> și <i>n</i>, care reprezintă numărul de linii și coloane al unui tablou (ambele mai mici ca 100). În următoarele <i>m</i> rânduri sunt scrise câte <i>n</i> numere reale – elementele fiecărei linii.</p> <pre>Var a:array [1..100,1..100] of real; i,j,m,n : integer; f:text; Begin Assign (f, 'tablou.txt'); reset(f); readln(f, m, n);</pre>		

```

For i:=1 to m do
begin   For j:=1 to n do read(f,a[I,j]);
        readln(f);
end; ...
tablou.txt
3 5
2 4 78 43 121
123 43 546 76 8
8 5 2 45 312

```

Exemplu 2: Fișierul *date.txt* conține mai multe numere întregi, care sunt scrise în mai multe rânduri. De citit datele din fișier și de afișat la ecran aceste numere.

```

date.txt
23 5 456 6 8 9 80 65 432
524 78 143 121
123
43 546
76 8
18 55 26 445 12

```

Atragem atenția că în diferite rânduri a fișierului sunt scrise un număr de valori diferite.

```

Var a: integer; f:text;
Begin Assign (f,'date.txt'); reset(f);
While not eof(f) do
Begin While not eoln(f) do
        Begin read(f,a); write(a:8); end;
        readln(f); writeln;
end; ...

```

Fișierul tipizat poate fi perceput ca un tablou unidimensional cu indicii 0,1,... Toate elementele fișierului au un indice. În timpul lucrului cu acest tip de fișier pot fi folosite următoarele:

Instrucțiunea `seek(f,n)` – cursorul fișierului trece la componenta cu indicele n.

`Filesize(f)` – returnează numărul de componente a fișierului.

`Filepos(f)` – returnează poziția cursorului în fișier.

C

```

FILE *nume_file;
ex : FILE *f;

```

Pentru deschiderea fișierului se va utiliza funcția `fopen`. Mod de utilizare :

```

FILE *fopen(adresa_file, mod_de_deschidere);

```

Deschide fișierul cu numele dat pentru acces de tip `mod_de_deschidere`. Returnează pointer la fișier sau NULL dacă fișierul nu poate fi deschis, valoarea returnată este memorată în variabila `fișier`, care a fost declarată pentru accesarea lui.

`adresa_file` indică numele și adresa fișierului. Pentru a deschide fișierul

azi.txt în calitate de adresă vom indica "*azi.txt*", dacă fișierul este pe discul c în mapa temp atunci vom scrie "*C: \\temp\\azi.txt*".

Parametrul *mod_de_deschidere* este un șir de caractere care poate avea următoarele valori :

"r" - readonly , este permisă doar citirea dintr-un fișier existent;

"w" - write, creează un nou fișier, sau dacă există deja, distruge vechiul conținut;

"a" - append, deschide pentru scriere un fișier existent (scrierea se va face în continuarea informației deja existente în fișier, deci pointerul de acces se plasează la sfârșitul fișierului);

"+" - permite scrierea și citirea - actualizare (ex: "r+", "w+", "a+"). O citire nu poate fi direct urmata de o scriere și reciproc. Întâi trebuie re poziționat cursorul de acces printr-un apel la *fseek*.

"b" - specifică fișier de tip binar.

Funcții de prelucrare a fișierelor

- **int** *feof(FILE *fp)*; returnează o valoare nenulă dacă s-a întâlnit sfârșitul de fișier la ultima operație de intrare și 0 în caz contrar;

- **int** *fclose(FILE *fp)*; unde *fp* este pointerul returnat de funcția *fopen*. Prin închiderea fișierului se eliberează zona tampon alocată acestuia. Funcția returnează valoarea 0 dacă operația de închidere s-a efectuat cu succes;

- **int** *fsetpos(FILE *fp, long int *poz)*; atribuie indicatorului valoarea variabilei indicată prin pointerul *poz* și returnează valoarea 0 în caz de succes;

- **int** *fseek(FILE *fp, long offset, int whence)*; re poziționează indicatorul fișierului la valoarea *whence+offset*; *whence* poate avea următoarele valori:

SEEK_SET = 0 - început de fișier;

SEEK_CUR = 1 - poziție curentă;

SEEK_END = 2 - sfârșit de fișier.

- **int** *getc(FILE *fp)*; returnează următorul caracter citit din fișierul *fp* sau EOF dacă se întâlnește sfârșitul fișierului;

- **int** *putc(int c, FILE *fp)*; înscrie caracterul *c* în fișierul *fp*, în caz de eroare returnează EOF;

- **char** **fgets(char *s, int n, FILE *fp)*; citește maxim *n-1* caractere din fișierul *fp* sau până la '\n' inclusiv, le depune în *s*, adaugă la sfârșit '\0' și returnează adresa șirului *s*, în caz de eroare întoarce valoarea NULL;

- **int** *fputs(char *s, FILE *fp)*; scrie șirul *s* în fișier, fără caracterul '\0', în caz de eroare întoarce valoarea EOF;

	<ul style="list-style-type: none"> ▪ <code>int fscanf(FILE *fp, char *format [, adresa, ...]);</code> citește date din fișierul <code>fp</code>, conform formatului specificat (specific funcției <code>fscanf</code>); ▪ <code>int fprintf(FILE *fp, char *format [, argum...]);</code> scrie date, conform formatului (specific funcției <code>fprintf</code>) în fișierul <code>fp</code>, returnează numărul de octeți transferați sau EOF în caz de eroare.
C++	<p>Un fișier este văzut în C++ ca un obiect, deci ca o variabilă de un tip clasă. Se pot folosi 3 clase predefinite în biblioteca <code>fstream.h</code> :</p> <ul style="list-style-type: none"> ▪ <code>fstream</code> pentru fișiere ce pot fi folosite în citire sau în scriere; ▪ <code>ifstream</code> pentru fișiere din care este permisă doar citirea; ▪ <code>ofstream</code> pentru fișiere în care este permisă doar scrierea. <p>Declararea unei variabile de tip fișier :</p> <pre>nume_clasa nume_file;</pre> <p>Deschiderea fișierului se efectuează prin intermediul metodei <code>open</code>. Mod de utilizare :</p> <pre>nume_file.open(adresa_file, mod_de_deschidere);</pre> <p>Argumentul <code>mod_de_deschidere</code> este opțional pentru streamuri de intrare <code>ifstream</code>, pentru care valoarea implicită este <code>ios::in</code>, și pentru streamuri de ieșire <code>ofstream</code>, pentru care valoarea implicită este <code>ios::out</code>. Pentru streamuri de intrare/ieșire <code>fstream</code>, argumentul <code>mod_de_deschidere</code> trebuie să aibă una din valorile definite în clasa <code>ios</code> :</p> <ul style="list-style-type: none"> <code>ios::in</code> este permisă doar citirea dintr-un fișier existent; <code>ios::out</code> creează un nou fișier, sau dacă există deja, distruge vechiul conținut; <code>ios::app</code> deschide un fișier pentru adăugare, pointerul de acces se plasează la sfârșit; <code>ios::ate</code> deschide un fișier existent pentru citire sau scriere, pointerul de acces se plasează la sfârșit; <code>ios::trunc</code> deschide un fișier și ștergere vechiul conținut; <code>ios::binary</code> specifică fișier de tip binar. <p>În argumentul <code>mod_de_deschidere</code> se pot combina prin operatorul OR (SAU) două sau mai multe din aceste valori definite.</p> <p>Implicit, fișierele se deschid în mod <code>text</code>. Valoarea <code>ios::binary</code> determină deschiderea în mod binar a fișierului. Orice fișier poate fi deschis în mod <code>text</code> sau mod binar, indiferent de felul în care au fost formate datele.</p> <p>Scrierea datelor în fișier se realizează prin intermediul operatorilor de ieșire:</p> <pre>Nume_file<<flux1<<flux2<<...<<fluxn;</pre> <p>Citirea datelor din fișier se realizează prin intermediul operatorilor de intrare:</p> <pre>Nume_file >>var1>>var2>>...>>varn;</pre>

Valoarea argumentului acces determină tipul de acces la fișier.

- `int eof()`; returnează nenul dacă este sfârșit de fișier;
- `int bad()`; returnează nenul dacă s-a produs o eroare;
- `int fail()`; returnează nenul dacă operația nu a reușit.
- `get(ch)`; extrage un caracter în `ch`;
- `getline(str, MAX, [DELIM])`; extrage până la `MAX` caractere din `str` sau până la caracterul `DELIM` (`'\0'` sau `'\n'`);
- `read(str, MAX)`; extrage până la `MAX` caractere în `str` sau până la EOF;
- `seekg(positie)` setează distanța (în bytes) a pointerului de fișier față de începutul fișierului;
 - `seekg(positie, seek_dir)` setează distanța (în bytes) a pointerului de fișier față de poziția specificată `seek_dir`, care poate lua următoarele valori:
 - `ios::beg` - început de fișier
 - `ios::cur` - poziție curentă
 - `ios::end` - sfârșit de fișier
- `tellg(pos)` returnează poziția (în bytes) a pointerului de fișier față de începutul fișierului;
- `write(str, SIZE)` inserează `SIZE` caractere din vectorul `str` în stream.
- `tellp()` returnează poziția pointerului de fișier în bytes.

Probleme pentru rezolvare independentă

1) Fișierul *date.in* conține 3 valori separate printr-un singur spațiu, care reprezintă vârsta, înălțimea și genul unei persoane. Să se scrie un program care determină greutatea ideală a unei persoane cunoscând înălțimea, vârsta și genul persoanei (f sau m). Formulele de calcul sunt:

$$G_{\text{masculin}} = 50 + 0.75 * (\text{inaltime} - 150) + (\text{varsta} - 20) / 4,$$

$$G_{\text{feminin}} = G_{\text{masculin}} - 10,$$

unde înălțimea este exprimată în cm și vârsta în ani.. La ecran se va afișa greutatea ideală. Toate datele despre persoană (vârsta, înălțimea, genul, greutatea ideală) se vor scrie în fișierul *date.out*.

2) Fișierul *date.in* conține 3 numere întregi separate printr-un singur spațiu. Să se elaboreze un program care va afișa aceste numere unul sub altul, afișând în dreptul fiecăruia unul dintre cuvintele PAR sau IMPAR. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

3) Fișierul *date.in* conține 3 numere întregi distincte separate printr-un singur spațiu. Să se elaboreze un program care va specifica care dintre numere are valoare

maximală, minimală și care nu este nici minim nici maxim. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*.

4) Fișierul *date.in* conține 12 numere întregi separate printr-un singur spațiu care reprezintă temperaturile medii ale lunilor unui an, ca numere întregi. Să se afișeze cu două zecimale media anuală a temperaturilor pozitive și a celor negative. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*.

5) Fișierul *date.in* conține n numere întregi separate printr-un singur spațiu. Primul număr reprezintă numărul de numere din fișier. Datele din fișier reprezintă mediile a n elevi, ca numere reale. Să se afișeze cea mai mare și cea mai mică medie. Să se verifice dacă sunt corigenți. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*
Exemplu : Date de intrare 4 9.50 4.25 9.66 6.33 Date de ieșire max=9.66 min=4.25
1 corigent.

6) Fișierul *date.in* conține n numere întregi separate printr-un singur spațiu. Primul număr reprezintă numărul de numere din fișier. Să se afișeze numărul minimal, maximal, numărul de elemente pare, impare și media numerelor cu două zecimale. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

7) Fișierul *date.in* conține elementele unei matrice. În prima linie sunt scrise numărul de linii și de coloane separate printr-un spațiu. Pe următoarele linii sunt scrise elementele matricei. Elaborați un program care va afișa la ecran și va scrie în fișierul *date.out* elementele matricei și pozițiile pe care se află valoarea maximală.

8) Fișierul *date.in* conține elementele unei matrice. În prima linie sunt scrise numărul de linii și de coloane separate printr-un spațiu. Pe următoarele linii sunt scrise elementele matricei. Elaborați un program care va afișa la ecran și va scrie în fișierul *date.out* elementele matricei și suma de pe fiecare linie.

9) Fișierul *date.in* conține elementele unei matrice. În prima linie sunt scrise numărul de linii și de coloane separate printr-un spațiu. Pe următoarele linii sunt scrise elementele matricei. Elaborați un program care va afișa la ecran și va scrie în fișierul *date.out* elementele matricei și minimul de pe fiecare coloană.

10) Fișierul *date.in* conține un șir de caractere. Elaborați un program care va determina numărul de majuscule din șir. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

11) Fișierul *date.in* conține un șir de caractere. De la tastatură se citește un șir de caractere. Elaborați un program care va înlocui cifrele din șir cu semnul +. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

12) Fișierul *date.in* conține un șir de caractere. De la tastatură se citește un șir de caractere. Elaborați un program care va afișa doar cifrele din șir. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

13) Fișierul *date.in* conține un șir de caractere. De la tastatură se citește un șir de caractere. Elaborați un program care va determina numărul de cifre din șir. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

14) Fișierul *date.in* conține un șir de caractere. De la tastatură se citește un șir de caractere. Elaborați un program care va înlocui vocalele din șir cu spațiu. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

15) Fișierul *date.in* conține un șir de caractere. De la tastatură se citește un șir de caractere. Elaborați un program care va determina numărul de caractere speciale (paranteze, operatori aritmetici) din șir. Rezultatul va fi afișat la ecran cât și în fișierul *date.out*

16) În fișierul *numere.in* sunt scrise mai multe numere întregi. Elaborați un program prin intermediul căruia se vor adăuga numerele pare din fișierul *numere.in* în fișierul *date.txt*.

17) Elaborați un program prin intermediul căruia va fi creat un fișier. Numele fișierului se va citi de la tastatură. În fișier vor fi scrise toate numerele pare mai mici decât $n(n < 10000)$, n se citește de la tastatură.

18) Elaborați un program prin intermediul căruia va fi creat un fișier. Numele fișierului se va citi de la tastatură. În fișier vor fi scrise toate numerele impare mai mici decât $n(n < 10000)$, n se citește de la tastatură.

19) Fișierul *numere.in* conține mai multe numere reale (cel puțin 2). Elaborați un program prin intermediul căruia în acest fișier se va adăuga media aritmetică a ultimelor 2 numere.

20) Fișierul *numere.in* conține mai multe numere reale (cel puțin 2). Elaborați un program prin intermediul căruia în acest fișier se va adăuga media aritmetică a numerelor din fișier.

21) În fișierul *date.in* este scris un număr întreg pozitiv mai mic decât 1000. Elaborați un program care va adăuga în fișier un triunghi:

a)	b)	c)
4	4	4
1	1234	1
12	123	21
123	12	321
1234	1	4321

22) În fișierul *date.in* sunt scrise mai multe cuvinte, fiecare cuvânt este separat printr-un singur spațiu. Elaborați un program prin intermediul căruia se va afișa la ecran conținutul fișierului, cel mai lung cuvânt și cel mai mare cuvânt.

§5 Metode de sortare

Procedura de sortare se aplică foarte larg în informatică.

Prin sortare vom înțelege operația de aranjare a elementelor într-o anumită ordine: crescătoare sau descrescătoare.

Existând mai multe metode de sortare în continuare vom menționa unele din ele. Admitem că avem declarațiile:

Varianta Pascal	Varianta C/C++
<pre>var a : array [1..100] of integer; x,i,j,n,k,l:integer; p:boolean;</pre>	<pre>int a[100],x, i,j,n,k,l.p;</pre>
Se cunosc: valoarea lui n – numărul de elemente al tabloului, și elementele tabloului a .	
Metoda de inserție	
<pre>For i:=2 to n do Begin x:=a[i]; j:=i-1; While (j<=1) and (a[j]>x) do Begin a[j+1]:=a[j]; j:=j-1; end; a[j+1]:=x; end;</pre>	<pre>for(i=1;i<n;i++){ x=a[i]; j=i-1; while(j<=0&&a[j]>x){ a[j+1]=a[j]; j--; } a[j+1]=x; }</pre>
Metoda bulelor	
<pre>k:=n; p:=false; while p=false do Begin p:=true; l:=k; i:=1; While i<=l-1 do Begin If a[i]>a[i+1] then begin x:=a[i]; a[i]:=a[i+1]; a[i+1]:=x; k:=i; p:=false; end; i:=i+1; end; end;</pre>	<pre>k=n; p=0; while (!p){ p=1; l=k; i:=0; while(i<=l-1){ if(a[i]>a[i+1]){ x=a[i]; a[i]=a[i+1]; a[i+1]=x; k:=i; p=0; } i++; }; };</pre>
Metoda de selecție	
<pre>i:=1; while i<=n-1 do Begin x:=a[i]; l:=i; j:=i+1; While j<=n do Begin if x>a[j] then begin x:=a[j]; l:=j; end; j:=j+1; end; k:=a[i]; a[i]:=a[l]; a[l]:=k;</pre>	<pre>i=0; while (i<n-1){ x=a[i]; l=i; j=i+1; while(j<n){ if (x>a[j]){x=a[j];l=j;} j++; }; k=a[i]; a[i]=a[l];</pre>

<pre> i:=i+1; end; sau for i:=1 to n-1 do begin x:=a[i]; l:=i; for j:=i+1 to n do if x>a[j] then begin x:=a[j]; l:=j; end; k:=a[i]; a[i]:=a[l]; a[l]:=k; end; </pre>	<pre> a[l]=k; i++; }; </pre>
<p>Sortarea prin distribuire</p> <p>Această metodă se folosește de obicei în cazul când elementele tabloului numeric sunt cuprinse într-un domeniu nu prea mare. Implementarea acestei metode se reduce la declararea unui tablou suplimentar de dimensiunile domeniului. În acest tablou inițial zerografiat, se adună câte o unitate în celula cu indicele egal cu valoarea elementului din tabloul inițial. Fie tabloul <i>a</i> – datele inițiale și <i>b</i> - cel cu rezultate, domeniul fiind de la <i>c</i> la <i>d</i>.</p>	
<pre> For i:=c to d do b[i]:=0; For i:=1 to n do begin t:=a[i]; b[t]:=b[t]+1; end; {aranjăm elementele în tabloul a} i:=1; For j:=c to d do For k:=1 to b[j] do begin a[i]:=j; i:=i+1; end; </pre>	<pre> for(i=c;i<d;i++) b[i]=0; for(i=0;i<n;i++){ t=a[i]; b[t]=b[t]+1; } {aranjăm elementele în tabloul a} i=0; for(j=c;j<d;j++) for(k=0;k<b[j];k++){ a[i]=j; i++; } </pre>

Probleme pentru rezolvare independentă

1) Fișierul *vector.in* conține *n* numere întregi separate printr-un singur spațiu. Primul număr reprezintă numărul de numere din fișier. Elaborați un program prin intermediul căruia la ecran se vor afișa numerele sortate crescător și descrescător, utilizând metoda bulelor, inserției și selecției.

2) Fișierul *caractere.in* conține mai multe caractere. Elaborați un program prin intermediul căruia la ecran se vor afișa caracterele sortate crescător și descrescător, metoda bulelor, inserției și selecției.

3) Fișierul *vector.in* conține *n* numere întregi separate printr-un singur spațiu. Primul număr reprezintă numărul de numere din fișier. Elaborați un program prin

intermediul căruia primele k numere vor fi sortate crescător, iar restul numerelor descrescător, k se citește de la tastatură.

4) Fișierul *vector.in* conține n numere întregi separate printr-un singur spațiu. Primul număr reprezintă numărul de numere din fișier. Elaborați un program prin intermediul căruia primele k numere vor fi sortate descrescător, iar restul numerelor crescător, k se citește de la tastatură.

5) Fișierul *cuvinte.in* conține mai multe cuvinte, separate printr-un singur spațiu. Elaborați un program prin intermediul căruia la ecran se vor afișa cuvintele aranjate descrescător/crescător.

6) Fișierul *matrice.in* conține elementele unei matrice. În prima linie sunt scrise numerele m și n – numărul de linii și de coloane separate printr-un spațiu. Pe următoarele n linii sunt scrise câte m elemente ale matricei. Elaborați un program prin intermediul căruia se vor sorta elementele fiecărei coloane în parte a matricei.

7) Fișierul *matrice.in* conține elementele unei matrice. În prima linie sunt scrise numărul de linii și de coloane separate printr-un spațiu. Pe următoarele linii sunt scrise elementele matricei. Elaborați un program prin intermediul căruia se vor sorta elementele fiecărei linii a matricei.

§6 Tipul de date articol

1) De la tastatură se citește numele și data nașterii a trei copii. Elaborați un program prin intermediul căruia la ecran se va afișa copilul cu vârsta cea mai mică, cea mai mare și cea mijlocie.

2) De la tastatură se citesc datele despre două variabile de tipul *data* (an, luna, zi). Elaborați un program prin intermediul căruia la ecran se va afișa suma și diferența dintre cele două variabile de tipul *data*. La ecran data va fi afișată corect ($0 \leq zi \leq 30$, $0 \leq luna \leq 12$)

3) De la tastatură se citesc datele despre două variabile de tipul *timp* (ore, min, sec). Elaborați un program prin intermediul căruia la ecran se va afișa suma și diferența dintre cele două variabile de tipul *timp*. La ecran timpul va fi afișată corect ($0 \leq sec \leq 59$, $0 \leq min \leq 59$).

4) Elaborați un program prin intermediul căruia:

A. De la tastatură se citesc coordonatele vârfurilor unui triunghi (3 vârfuri).

Determinați:

a) Dacă coordonatele introduse pot forma un triunghi;

b) În cazul în care triunghiul poate fi format:

-) se va afișa aria și perimetrul triunghiului;

-) tipul triunghiului: ascuțit, obtuz, drept.

B. De la tastatură se citesc datele despre două cercuri (coordoanatele centrului și raza). Determinați:

a) Pentru fiecare cerc aria și lungimea discului;

b) Poziția cercurilor (unul în interiorul altuia, tangente, au două puncte comune, nu au nici un punct comun (exclue cazul când un cerc este situat în interiorul altui cerc)).

Formule necesare:

Distanța dintre 2 puncte $A(x_1, y_1)$ și $B(x_2, y_2)$ $d_{AB} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Aria unui triunghi: $Aria = \sqrt{P(P-a)*(P-b)*(P-c)}$, unde $P = \frac{a+b+c}{2}$

5) Elaborați un program prin intermediul căruia se va crea tipul de date fracție.

De la tastatură se citesc datele despre două fracții, la ecran se va afișa:

a) ce mai mare și ce mai mică fracție;

b) suma fracțiilor;

- c) diferența fracțiilor;
- d) produsul fracțiilor;
- e) câtul fracțiilor;

Fracțiile se vor afișa doar sub formă de fracții ireductibile de forma a/b , de exemplu $3/4$.

6) Pe o suprafață plană sunt mai multe puncte, date prin coordonatele lor. Elaborați un program prin intermediul căruia se vor sorta crescător în dependență de distanța acestora de la centru axei de coordonate a sistemului cartezian.

Date de intrare. Fișierul puncte.in, care pe prima linie conține numărul de puncte din plan, iar pe următoarele linii sunt scrise coordonatele punctelor.

Date de ieșire. La ecran se va afișa punctele sortate, fiecare coordonată de punct fiind urmată de distanța acestuia de la centru.

7) Prin intermediul tipului de date **articol**, creați tipul de date telefon cu câmpurile: marca, culoare, anul fabricării, prețul. În baza acestui tip elaborați un program prin intermediul căruia se vor gestiona datele despre telefoanele mobile (minim 10 telefoane). Citirea datelor se va efectua din fișier. Programul va conține un meniu cu următoarele opțiuni:

- a) Afișarea tuturor telefoanelor;
- b) Afișarea telefoanelor cu cel mai mare an;
- c) Afișarea celor mai scumpe telefoane;
- d) Căutarea telefoanelor după marcă;
- e) Căutarea telefoanelor după preț, cu indicarea prețului minim și maxim;
- f) Sortarea telefoanelor descrescător după anul fabricării;
- g) Sortarea telefoanelor crescător după marcă.

§7 Recapitulare

1) Se consideră două tablouri $A[1..N]$ și $B[1..N]$ de numere reale, $N \leq 100$. Scrieți un program care numără cazurile cu următoarea proprietate:

$$A[i] < B[i]; \quad A[i] = B[i]; \quad A[i] > B[i].$$

Intrare: Numărul N și elementele tablourilor A și B se citesc de la tastatură.

Ieșire: La ecran se vor afișa trei rânduri, indicând proprietatea și numărul de situații respective.

2) Se consideră tabloul $A[1..N]$ de numere întregi, $N \leq 100$. Scrieți un program care calculează media aritmetică a numerelor de pe locurile pare și media aritmetică a numerelor de pe locurile impare.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa două valori.

3) Se consideră tabloul $A[1..N]$ de numere întregi, $N \leq 100$. Scrieți un program care calculează suma elementelor mai mici ca elementul maxim.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se va afișa suma cerută.

4) Se consideră tabloul $X[1..N]$ de numere întregi, $N \leq 100$. Scrieți un program care determină lungimea celei mai lungi secvențe de elemente consecutive ordonate crescător.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se va afișa mărimea celei mai lungi secvențe de elemente ordonate crescător și pozițiile începutului și sfârșitului ei.

5) Se consideră tabloul $A[1..N]$ de numere întregi, $N \leq 100$. Scrieți un program care efectuează deplasarea spre stânga cu o poziție a elementelor $A[2]$, $A[3]$, ..., $A[n]$. Primul element va fi înscris pe ultima poziție.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tabloului modificat.

6) Se consideră tabloul $A[1..N]$ de numere întregi, $N \leq 100$. Scrieți un program care efectuează deplasarea elementelor spre stânga cu M poziții. Primele M elemente vor fi înscrise pe ultimele M poziții.

Intrare: Numerele N , M și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tabloului modificat.

7) Se consideră tabloul $X[1..10]$ de numere întregi egale numai cu 0, 1 sau 2. Scrieți un program care aranjează pe primele locuri elementele egale cu 1, apoi cele egale cu 0, apoi cele egale cu 2.

Intrare: Elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se va afișa elementele tabloului modificat.

8) Se consideră tabloul $A[1..2*N]$ de numere naturale, unde N elemente sunt pare, iar N elemente sunt impare, $N \leq 50$. Scrieți un program care aranjează elementele astfel: pe locuri pare - elemente cu valori pare, pe locuri impare - elemente cu valori impare păstrând ordinea.

De exemplu, pentru tabloul: 4 5 6 7 8 9 se va obține: 5 4 7 6 9 8

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tabloului modificat.

9) Se consideră tabloul $X[1..N]$ de numere reale, $N \leq 100$. Scrieți un program care creează tabloul Y din elementele tabloului X , inserând între fiecare 2 elemente media lor aritmetică. De exemplu, având tabloul: 7 4.2 11 -1 20 se va obține tabloul: 7 5.6 4.2 7.6 11 5 -1 9.5 20

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: Elementele tabloului Y se vor afișa la ecran.

10) Se consideră tabloul $T[1..N, 1..M]$ cu numere reale, $N \leq 20$ și $M \leq 20$. Scrieți un program care interschimbă cu locurile elementul maxim de pe fiecare coloană cu elementul din prima linie a aceleiași coloane.

Intrare: Numerele N , M și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tabloului modificat

11) Se consideră două tablouri liniare $A[1..N]$ și $B[1..N]$, $N \leq 100$. Scrieți un program care modifică elementele acestor tablouri conform regulii: $A[i] := A[i] + B[i]$ și $B[i] := A[i] * B[i]$. De exemplu, dacă $A[i] = 7$ și $B[i] = 3$ atunci după modificare obținem valorile lui $A[i] = 10$ și $B[i] = 21$.

Intrare: Numărul N și elementele tablourilor se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tablourilor modificate.

12) Se consideră tabloul $X[1..N]$ de numere întregi nenule, $N \leq 100$. Scrieți un program care numără de câte ori se schimbă semnul numerelor din acest tablou. De exemplu: pentru tabloul 2 9 -5 -4 6 12 -8 5 32 1 sunt 4 alternări de semne.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se va afișa numărul de alternări de semne.

Capitolul 3

SUBPROGRAME

§1 Parametri formali de tip valoare

1) De la tastatură se citește un număr întreg n , $n < 100000$. Elaborați un program cu funcții prin intermediul căruia se va determina:

- a) Numărul de cifre;
- b) Numărul de cifre pare;
- c) Numărul de cifre impare;
- d) Suma cifrelor;
- e) Cifra maximă;
- f) Cifra minimă;
- g) Media aritmetică a cifrelor;
- h) Cifrele care se repetă de cel puțin două ori;
- i) Cifrele numărului separate printr-un sigur spațiu;
- j) Divizorii numărului;
- k) Inversul numărului;
- l) Mesajul PRIM, dacă numărul este număr prim;
- m) Cel mai mare număr posibil creat din aceste cifre.

2) De la tastatură se citesc două numere întregi. Elaborați un program cu funcții prin intermediul căruia se va determina:

- a) Suma numerelor;
- b) Produsul numerelor;
- c) Media aritmetică a numerelor;
- d) Cel mai mare divizor comun;
- e) Cel mai mic multiplu comun;
- f) Numărul minim;
- g) Numărul maxim;
- h) Suma numerelor în formatul $a+b=c$, dacă a și b reprezintă numerele citite;
- i) Produsul numerelor în formatul $a*b=c$, dacă a și b reprezintă numerele citite;
- j) Toți divizorii comuni;
- k) Cinci multipli comuni;
- l) Cifrele care se conțin în ambele numere;
- m) Cifrele care sunt în primul număr și nu sunt în al doilea număr;

- n) Va afișa mesajul PRIETENE, dacă numerele sunt prietene. Două numere se numesc prietene, dacă numărul de divizori este același.

3) De la tastatură se citesc trei numere întregi. Elaborați un program cu funcții prin intermediul căruia:

- a) Se va determina cel mai mare divizor comun al numerelor;
- b) Se va determina cel mai mic multiplu comun al numerelor;
- c) Se va determina valoarea maximă;
- d) Se va determina valoarea minimă;
- e) Se vor afișa toți divizorii comuni;
- f) Se vor cei mai mici trei multipli comuni.
- g) Se va verifica dacă numerele citite pot fi lungimile laturilor unui triunghi;
 - g.1. Se va determina aria triunghiului, dacă valorile citite pot forma un triunghi;
 - g.2. Se va determina perimetrul triunghiului, dacă valorile citite pot forma un triunghi;
- h) Se va afișa soluțiile reale ale ecuației $ax^2+bx+c=0$, dacă a, b, c reprezintă coeficienții ecuației.

§2 Parametri formali de tip variabilă

1) De la tastatură se citesc două numere întregi. Pentru fiecare dintre următoarele cazuri să se creeze câte o procedură, prin intermediul căreia se va determina:

- a) Suma numerelor;
- b) Produsul numerelor;
- c) Media aritmetică a numerelor;
- d) Cel mai mare divizor comun;
- e) Cel mai mic multiplu comun;
- f) Numărul minim;
- g) Numărul maxim.

2) Se consideră tipul de date *timp (ora:min:sec)*. Elaborați un program prin intermediul căruia se va crea câte o procedură pentru:

- a) Citirea timpului de la tastatură;
- b) Afișarea timpului la ecran, în formatul *ora:min:sec*, dacă $0 \leq sec \leq 59$ și $0 \leq min \leq 59$.
- c) Determinarea sumei dintre cele două variabile de tipul *timp*;
- d) Determinarea diferenței dintre cele două variabile de tipul *timp*;
- e) Determinarea valorii maxime;
- f) Determinarea valorii minime;

3) Se consideră tipul de date *data (an:luna:zi)*. Elaborați un program prin intermediul căruia se va crea câte o procedură pentru:

- a) Citirea datei de la tastatură;
- b) Afișarea datei la ecran, în formatul *an:luna:zi*, dacă $1 \leq zi \leq 30$ și $1 \leq luna \leq 12$.
- c) Determinarea sumei dintre cele două variabile de tipul *data*;
- d) Determinarea diferenței dintre cele două variabile de tipul *data*;
- e) Determinarea valorii maxime;
- f) Determinarea valorii minime;

4) Se consideră tipul de date *polinom (ax^2+bx+c)*. Elaborați un program prin intermediul căruia se va crea câte o procedură pentru:

- a) Citirea coeficienților polinomului de la tastatură;
- b) Afișarea polinomului la ecran, în formatul ax^2+bx+c , dacă a, b, c reprezintă coeficienții polinomului.
- c) Determinarea sumei dintre cele două polinoame;
- d) Determinarea diferenței dintre cele două polinoame;
- e) Determinarea polinomului maxim;

- f) Determinarea polinomului minim;
- g) Determinarea soluțiilor reale ale polinomului.

5) Se consideră un triunghi cu lungimile laturilor egale cu a , b , c , - numere reale. Pentru fiecare dintre următoarele cazuri să se creeze câte o procedură, prin intermediul căreia:

- a) Se va citi de la tastatură lungimile laturilor;
- b) Se va determina dacă valorile a, b și c au fost introduse corect (pot forma un triunghi), în caz contrar se va repeta operația de citire, până când valorile nu vor fi introduse corect;
- c) Se va determina aria și perimetrul triunghiului.
- d) Se va afișa tipul triunghiului (ascuțit, drept, obtuz).

§3 Tablouri și funcții

1) Se consideră un tablou unidimensional. Elaborați un program cu funcții/proceduri prin intermediul cărora:

- a) De la tastatură se vor citi elementele tabloului;
- b) Se vor afișa elementele tabloului la ecran;
- c) Se vor sorta elementele crescător;
- d) Se vor inversa elementele tabloului;
- e) Va fi determinat elementul maximal;
- f) Va fi determinat elementul minimal;
- g) Se va determina numărul de elemente maxime;
- h) Se va determina numărul de elemente minime.

2) Se consideră două tablouri cu numere întregi, a și b. Elaborați un program utilizarea funcțiilor/procedurilor prin intermediul cărora:

- a) De la tastatură se citesc elementele tablourilor;
- b) Se afișează la ecran elementele tablourilor;
- c) Va determina suma elementelor fiecărui tablou;
- d) Va determina diferența elementelor fiecărui tablou;
- e) Va crea un nou tablou care va conține toate elementele tabloului a și b;
- f) Va crea un nou tablou format doar din elementele comune ambelor tablouri.

3) Se consideră un tablou bidimensional cu numere întregi. Elaborați un program cu funcții/proceduri prin intermediul cărora:

- a) De la tastatură se vor citi elementele tabloului;
- b) Se vor afișa elementele tabloului la ecran;
- c) Va fi determinat elementul maximal;
- d) Va fi determinat elementul minimal;
- e) Se va determina numărul de elemente maxime;
- f) Se va determina numărul de elemente minime;
- g) Se va afișa sumele pe linii;
- h) Se va afișa sumele pe coloane;
- i) Se vor sorta elementele crescător pe linii;
- j) Va determina suma elementelor de pe *perimetrul* tabloului.

4) Se consideră un șir de caractere. Elaborați un program cu funcții/proceduri, prin intermediul căruia:

- a) Va determina dacă șirul este palindrom;
- b) Va afișa doar vocalele din șir;

- c) Va afișa doar cifrele din șir;
- d) Va determina numărul de cuvinte din șir;
- e) Va sorta crescător elementele șirului;
- f) Va inversa șirul;
- g) Va determina cuvântul maxim;
- h) Va determina cuvântul de lungime maximă;
- i) Va inversa fiecare cuvânt din șir.

5) Elaborați un program prin intermediul căruia vor fi create proceduri pentru conversia numerelor în șir de caractere și reciproc:

- a) Procedură pentru transformarea șirului în număr întreg, dacă șirul conține doar cifre;
- b) Procedură pentru transformarea unui număr întreg în șir;
- c) Procedură pentru transformarea șirului în număr real, dacă șirul conține doar cifre și caracterul *punct*;
- d) Procedură pentru transformarea unui real în șir.

§4 Subprograme recursive

1) De la tastatură se citește un număr întreg. Elaborați un subprogram recursiv, prin intermediul căruia se va determina valoarea expresiilor:

- a) $1+3+5+\dots+2n-1$;
- b) $2+4+6+\dots+2n$;
- c) $2+5+8+\dots+3n-1$;
- d) $1+5+9+\dots+4n-3$;
- e) $\frac{1}{2} + \frac{1}{7} + \frac{1}{12} + \dots + \frac{1}{5n-3}$;
- f) $\frac{1}{3} + \frac{1}{8} + \frac{1}{13} + \dots + \frac{1}{5n-2}$;
- g) $-\frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \frac{1}{16} - \dots + (-1)^n \frac{1}{2^n}$.

2) Se consideră un tablou unidimensional cu numere întregi. Elaborați un subprogram recursiv prin intermediul căruia:

- a) De la tastatură se citesc elementele acestui tablou unidimensional;
- b) Se vor afișa elementele tabloului în ordinea citirii lor;
- c) Se vor afișa elementele tabloului în ordine inversă a citirii lor;
- d) Va determina suma elementelor;
- e) Va determina suma elementelor pare;
- f) Va determina suma elementelor impare;
- g) Va afișa elementele pare;
- h) Va afișa elementele impare;
- i) Va verifica existența numărului k în tablou (k se introduce de la tastatură);
- j) Va verifica existența numerelor pare în tablou.

3) Elaborați un subprogram recursiv prin intermediul căruia:

- a) Se va determina produsul dintre două numere întregi prin adunări repetate;
- b) Se va determina câtul dintre două numere întregi prin scăderi repetate;
- c) Se va determina restul împărțirii dintre două numere întregi prin scăderi repetate;

4) Elaborați un subprogram recursiv, prin intermediul căruia se va determina dacă un număr întreg, este sau nu număr prim.

1) Se consideră tabloul $A[I..N, I..N]$ de numere întregi, unde $N \leq 20$. Scrieți un program care va aduna la fiecare element al tabloului valoarea elementului maxim de pe diagonala principală.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tabloului modificat.

2) Se consideră tabloul $A[I..N, I..N]$ de numere întregi, unde $N \leq 20$. Scrieți un program care calculează suma elementelor care nu se află pe diagonala secundară.

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se va afișa suma calculată.

3) Se consideră un tablou pătrat (numărul de linii este egal cu numărului de coloane) elementele căruia sunt numere întregi. Scrieți un program care modifică tabloul astfel că toate elementele acestea se măresc cu valoarea elementului situat pe diagonala secundară și în aceeași linie.

Intrare: Fișierul text **TABEL.IN** organizat astfel: prima linie al fișierului conține un număr pozitiv întreg $1 < N \leq 10$; celelalte N linii conțin câte N valori numerice separate prin spații

Ieșire: Fișierul text **TABEL.OUT** conține N linii a câte N valori separate prin spații. *De exemplu:*

TABEL.IN	TABEL.OUT
4	7 10 13 10
2 5 8 5	9 4 6 0
6 1 3 -3	10 4 7 -3
8 2 5 -5	22 6 11 12
11 -5 0 1	

4) Se consideră un tablou pătrat (numărul de linii este egal cu numărul de coloane) completat cu numere întregi. Scrieți un program care modifică tabloul astfel că toate elementele din fiecare linie se măresc cu valoarea elementului situat în prima coloană și în aceeași linie.

Intrare: Fișierul text **TABEL.IN** organizat astfel: prima linie al fișierului conține un număr pozitiv întreg $1 < N \leq 10$; celelalte N linii conțin câte N valori numerice separate prin spații

Ieșire: Fișierul text **TABEL.OUT** conține N linii a câte N valori separate prin spații

De exemplu:

TABEL.IN	TABEL.OUT
4	4 7 10 7
2 5 8 5	12 7 9 3
6 1 3 -3	16 10 13 3
8 2 5 -5	22 6 11 12
11 -5 0 1	

5) Se consideră un tablou pătrat (numărul de linii este egal cu numărul de coloane) completat cu numere întregi. Scrieți un program care modifică tabloul astfel că toate elementele de deasupra diagonalei principale se măresc cu valoarea elementului situat în prima linie și în aceeași coloană cu elementul modificat.

Intrare: Fișierul text **TABEL.IN** organizat astfel: prima linie al fișierului conține un număr pozitiv întreg $1 < N \leq 10$; celelalte N linii conțin câte N valori numerice separate prin spații

Ieșire: Fișierul text **TABEL.OUT** conține N linii a câte N valori separate prin spații. *De exemplu:*

TABEL.IN	TABEL.OUT
4	2 10 16 10
2 5 8 5	6 1 11 2
6 1 3 -3	8 2 5 0
8 2 5 -5	11 -5 0 1
11 -5 0 1	

6) Se consideră tabloul liniar $T[1..N]$ de tip întreg, $N \leq 100$. Scrieți un program care înscrie numerele nenule la începutul tabloului (păstrând ordinea lor) iar zerourile le plasează la finele tabloului. Restricție: se interzice utilizarea altor tablouri!

Intrare: Numărul N și elementele tabloului se citesc de la tastatură.

Ieșire: La ecran se vor afișa elementele tabloului modificat

7) Fișierul cu acces direct **DATE.DAT** conține $3*n$ numere reale (n triplete). Scrieți un program care modifică fișierul astfel: fiecare al doilea element din triplete consecutive se înlocuiește cu media aritmetică a vecinilor lui. De exemplu, dacă **DATE.DAT** conținea 4 7 6 8 2 10 1 5 7 atunci după îndeplinirea programului va conține 4 5 6 8 9 10 1 4 7.

Intrare: Fișierul **DATE.DAT**

Ieșire: Fișierul **DATE.DAT** modificat.

8) Fișierul text **ELEVI.IN** conține datele referitoare la examenele de admitere a cel mult 200 concurenți. Fiecărui elev îi corespund câte două linii din acest fișier, pe

prima aflându-se numele și prenumele său, iar pe a doua media obținută la examene. Scrieți un program care scrie în fișierul **ELEVI.OUT** lista elevilor admiși (care au media nu mai mică decât 5) ordonată descrescător după medii.

Intrare: Fișierul text **ELEVI.IN**

Ieșire: Fișierul text **ELEVI.OUT**, ce conține pe fiecare linie numele și prenumele elevilor admiși, separate printr-un spațiu.

9) În fișierul text **ELEVI.INT** se conțin date despre toți elevii absolvenți din Republica Moldova. În fiecare rând sunt înscrise datele despre o singură persoană astfel: nume, prenume, sex și înălțimea, despărțite printr-un spațiu. Din aceste date se va crea o listă simplu înlănțuită ordonată alfabetic, după nume. Parcurgeți lista în scopul calculării și afișării la ecran a înălțimii medii a fetelor și a băieților.

Intrare: Fișierul text **ELEVI.INT**, unde sexul este indicat prin *f* pentru fete și *b* pentru băieți.

Ieșire: La ecran se vor afișa mesajul respectiv și mediile cerute.

10) Se numesc numere „bine ordonate” acele numere care au cifrele în ordine strict crescătoare sau strict descrescătoare (de exemplu 7532 și 2589 sunt „bine ordonate”).

Scrieți un program care citește din fișierul text **DATE.IN** câte un număr de 4 cifre din fiecare linie, și creează alt fișier text **DATE.OUT** unde înscrie numerele „bine ordonate”. La ecran se va afișa numărul lor.

Intrare: Fișierul text **DATE.IN**

Ieșire: Fișierul text **DATE.OUT** La ecran se va afișa numărul de numere „bine ordonate”.

11) Se consideră un fișier text, **INTRARE.TXT**, care conține cuvinte separate prin câte un singur spațiu. Să se creeze un fișier text, **IESIRE.TXT**, care conține cuvintele textului în ordine alfabetică și numărul lor de apariții. Cuvintele textului apar o singură dată în fișierul de ieșire.

Intrare: Fișierul text **INTRARE.TXT**

Ieșire: Fișierul text **IESIRE.TXT**

Exemplu:

Pentru fișierul **INTRARE.TXT**:

A fost odata ca-n povesti

A fost ca niciodata

Din rude mari imparatesti

O prea frumoasa fata

fișierul **IESIRE.TXT** va fi:

A 2

Din 1

O 1
ca 1
ca-n 1
fata 1
fost 2
frumoasa 1
imparatesti 1
mari 1
niciodata 1
odata 1
povesti 1
prea 1
rude 1

12) Se consideră un fișier text *TEST.IN* alcătuit din mai multe linii. Scrieți un program care să afișeze pe ecran toate literele alfabetului englez, folosite în acest fișier, împreună cu codul lor ASCII și frecvența lor de apariție. Afișați, de asemenea, codurile ASCII ale caracterelor care apar în text cel mai frecvent.

Intrare: Fișierul text *TEST.IN*.

Ieșire: La ecran se vor afișa datele cerute.

De exemplu, pentru fișierul de intrare:

aBc bAaccaa
bc ac

pe ecran se va afișa:

A	65	1
B	66	1
a	97	5
b	98	2
c	99	5

Caracterele care apar in text cu frecventa maximă 5 sunt:

Caracterul cu codul ASCII 97

Caracterul cu codul ASCII 99

13) Se consideră fișierul text *CUVINTE.TXT* ce conține mai multe cuvinte formate din litere minuscule ale alfabetului latin. Scrieți un program care înscrie în fișierul *CUVINTE.OUT*, în ordinea alfabetică literele distincte din componența fișierului *CUVINTE.TXT* și numărul respectiv de apariții al fiecărei litere. Rezultatul se înscrie în fiecare rând câte o literă și numărul de apariții, separate printr-un spațiu.

14) Se cunoaște că fișierul text *INPUT.TXT* este alcătuit din numărul par de linii și conține în linii impare numele elevilor, dar în cele pare – notele acestora la o lucrare de control. Numărul maxim al elevilor în clasă – 40. Scrieți un program care citește fișierul dat și creează un fișier text *SORT.TXT* care conține în fiecare rând numele elevului și nota în așa fel ca liniile să fie sortate în ordinea descrescătoare a numelui.

Capitolul 4

STRUCTURI DINAMICE

Descriere	Funcția/Procedura
Alocarea memoriei în heap pentru o variabilă referință legată și încărcarea adresei în argumentul p.	<i>New(var p:POINTER);</i>
Eliberarea memoriei alocate în heap pentru o variabilă referință legată.	<i>Dispose(var p:POINTER);</i>
Lungimea celui mai mare bloc continuu existent în heap	<i>MaxAvail:LONGINT;</i>
Lungimea totală a spațiului de memorie disponibil în heap.	<i>MemAvail:LONGINT;</i>
Alocarea memoriei în heap pentru o variabilă dinamică de mărime l și încărcarea adresei ei în argumentul p.	<i>GetMem(VAR p:POINTER; l:WORD);</i>
Eliberarea memoriei alocate în heap pentru o variabilă dinamică de mărime l și de adresa memorată în p.	<i>FreeMem(VAR p:POINTER; l:WORD);</i>
Memorarea adresei vârfului heap-ului (HeapPtr) în argumentul p.	<i>Mark(VAR p:POINTER);</i>
Eliberarea memoriei alocate variabilelor dinamice prin apelul precedent al procedurii Mark (depune p în HeapPtr).	<i>Release(VAR p:POINTER);</i>

1) Prin intermediul structurilor dinamice de date, creați tipul de date telefon cu câmpurile: marca, culoare, anul fabricării, prețul. În baza acestui tip elaborați un program prin intermediul căruia se va gestiona datele despre telefoanele mobile (minim 10 telefoane). Citirea datelor se va efectua din fișier. Programul va conține un meniu cu următoarele opțiuni:

- a) Afișarea tuturor telefoanelor;
- b) Afișarea celor mai noi telefoane;
- c) Afișarea celor mai scumpe telefoane;
- d) Căutarea telefoanelor după marcă;
- e) Căutarea telefoanelor după preț, cu indicarea prețului minim și maxim;
- f) Sortarea telefoanelor descrescător după anul fabricării;
- g) Sortarea telefoanelor crescător după marcă.
- h) Inserare datelor despre un telefon nou;
- i) Excluderea unui telefon.

2) De la tastatură se citesc cuvinte. Să se creeze o listă simplu înlănțuită ordonată alfabetic, care conține în celule cuvintele distincte și frecvența lor de apariție. Se va afișa conținutul listei în ordine alfabetică

3) De la tastatură se citesc mai multe numere întregi. Să se creeze o listă ordonată crescător.

4) De la tastatură se citește numărul n și numele a n copii. Să se simuleze următorul joc: cei n copii stau într-un cerc. Începând cu un anumit copil, se numără copiii în sensul acelor de ceasornic. Fiecare al n -lea copil iese din cerc. Câștigă ultimul copil rămas în joc.

5) Se consideră un depou de locomotive cu o singură intrare și cu o singură linie de cale ferată, care poate cuprinde oricâte locomotive. Să se scrie programul care realizează dispecerizarea locomotivelor din depou. Programul prelucrează comenzi de intrare în depou a unei locomotive, de ieșire din depou a unei locomotive și de afișare a locomotivelor din depou.

6) De la tastatură se citesc cuvinte (șiruri de caractere). Să se scrie un program care creează un arbore de căutare, care conține în noduri cuvintele și frecvența lor de apariție. Să se afișeze apoi cuvintele în ordine lexicografică crescătoare și frecvența lor de apariție.

7) Informațiile pentru medicamentele unei farmacii sunt: nume medicament, preț, cantitate, data primirii, data expirării. Evidența medicamentelor se ține cu un program care are drept structură de date un arbore de căutare după nume medicament. Să se scrie programul care execută următoarele operații:

- a) creează arborele de căutare;
- b) caută un nod după câmpul nume medicament și actualizează câmpurile de informație;
- c) tipărește medicamentele în ordine lexicografică;
- d) elimină un nod identificat prin nume medicament;
- e) creează un arbore de căutare cu medicamentele care au data de expirare mai veche decât o dată specificată de la terminal.

8) Se consideră o listă unidirecțională câmpul informațional al fiecărei celule conține un număr real. Scrieți declarațiile necesare și fragmentul de program care ar adăuga între oricare două celule vecine a listei inițiale celule noi câmpul informațional al cărora va fi media aritmetică a numerelor din celulele vecine.

Ieșire: La ecran se afișează valorile câmpurilor informaționale ale celulelor listei modificate.

Capitolul 5

TEHNICI DE PROGRAMARE

§1 Metoda trierii

1) Se consideră numerele naturale din mulțimea $\{1, 2, 3, \dots, n\}$, $n < 100000$, n se citește din fișier. Să se determine toate elementele acestei mulțimi, pentru care suma cifrelor este un număr impar, divizibil cu trei;

2) Fișierul *date.in* conține mai multe caractere, să se afișeze cea mai lungă secvență de caractere identice. Dacă sunt mai multe secvențe, se va afișa prima. Exemplu: text: aabsssdadaaab se va afișa sss.

3) Se consideră numerele naturale din mulțimea $\{1, 2, 3, \dots, n\}$, $n < 100000$, n se citește din fișier. Să se determine toate elementele acestei mulțimi, pentru care suma cifrelor este divizibilă cu a și cu b , a și b se citesc din fișier.

4) Se consideră mulțimea $P = \{P_1, P_2, \dots, P_n\}$ formată din n puncte ($2 \leq n \leq 30$) pe un plan euclidian. Fișierul *date.in* conține în prima linie numărul n , care reprezintă numărul de puncte din plan pe liniile următoare se regăsesc coordonatele punctelor. Fiecare punct P_j este definit prin coordonatele sale x_j, y_j . Elaborați un program care afișează la ecran coordonatele punctelor P_a, P_b distanța dintre care este maximă. Distanța dintre punctele P_j, P_m se calculează cu ajutorul formulei:

$$d_{jm} = \sqrt{(x_j - x_m)^2 + (y_j - y_m)^2}.$$

5) Pe o suprafață plană sunt mai multe puncte, date prin coordonatele lor. Fișierul *puncte.in*, care pe prima linie conține numărul de puncte din plan, iar pe următoarele linii sunt scrise coordonatele punctelor. Elaborați un program prin intermediul căruia se determina numărul maximal de triunghiuri care pot fi formate din aceste puncte cu condiția că:

- Triunghiurile formate nu se intersectează;
- Triunghiurile formate sunt înscrise;
- Triunghiurile formate au un vârf comun.

La ecran se va afișa numărul de triunghiuri.

§2 Metoda Greedy

1) **Problema spectacolelor** Într-o sală într-o zi trebuie planificate n spectacole. Pentru fiecare spectacol se cunoaște intervalul în care se desfășoară: (st, sf). Se cere să se planifice un număr maxim de spectacole astfel încât să nu se suprapună.

2) **Problema rucsacului** O persoană are un rucsac cu care poate transporta o greutate maximă G . Persoana are la dispoziție n obiecte și cunoaște pentru fiecare obiect greutatea și câștigul care se obține în urma transportului său la destinație. Se cere să se precizeze ce obiecte trebuie să transporte persoana în așa fel încât câștigul să fie maxim. Persoana are posibilitatea să taie obiectele

3) **Problema bancomatelor** Scrieți un program, care afișează modalitatea de plată, a unui bancomat, folosind un număr minim de bancnote, a unei sume întregi S de lei ($S < 100000$). Plata se efectuează folosind bancnote cu valoarea 1, 5, 10, 50, 100, 200 și 500 de lei. Numărul de bancnote de fiecare valoare se citește din fișierul text **BANI.IN**, care conține 7 rânduri, în fiecare din care sunt indicate numărul de bancnote respectiv de 1, 5, 10, 50, 100, 200 și 500 de lei.

4) **Suma componentelor prime** Fie a o variabilă indexată, ale cărei componente $A(1), A(2), \dots, A(n)$ sunt numere naturale nenule. Să se determine suma componentelor care sunt numere prime. Atunci când un număr prim se repetă, el va fi luat în considerație o singură dată.

5) La un concurs de automobilism de la linia de start până la final sunt plasate n stații de benzină (la diferite distanțe). Având rezervorul plin, mașina unui concurent poate parcurge cel mult o distanță d (în kilometri). Concurentul dorește să se oprească de cât mai puține ori și desigur să parcurgă întreg drumul de la punctul de start la cel final. Descrieți o metodă eficientă pe care trebuie să o aplice concurentul și arătați că strategia respectivă conduce la o soluție optimă.

6) Pe o bandă magnetică sunt n programe. Pentru fiecare program i ($1 \leq i \leq n$) de lungime l_i se cunoaște probabilitatea p_i cu care poate fi apelat (cerut), $p_1 + p_2 + \dots + p_n = 1$. Pentru a citi un program trebuie să citim banda de la început (secvențial). În ce ordine să memorăm programele pentru a minimiza timpul mediu de citire a unui program oarecare?

1) **Labirintul** Se consideră planul unui labirint desenat pe o foaie de hârtie liniată în pătrățele. Pătrățelele hașurate reprezintă obstacolele, iar cele nehașurate - camerele și coridoarele labirintului.

În memoria calculatorului planul labirintului, este redat prin matricea $A = \|a_{ij}\|_{n \times m}$ $1 \leq n, m \leq 30$, unde $a_{ij} = 1$, dacă pătrățul este hașurat, $a_{ij} = 0$, în caz contrar. Călătorul se poate deplasa dintr-un pătrățel nehașurat numai atunci când ele au o latură comună. Elaborați un program care găsește, dacă există, un drum din pătrățul **B** în pătrățul **C**

	1	2	3	...	j	...	m
1							
2		B					
3							
...							
i							
...							
n							C

2) Elaborați un program care afișează la ecran toate modurile de a descompune un număr natural în sumă de k numere naturale distincte. Exemplu: pentru $n=9$ și $k=3$ soluțiile sunt: $1+2+6, 2+3+4, 1+3+5$ 3.

3) Se consideră n ($n < 50$) săculețe, numerotate respectiv: $1, 2, 3, \dots, n$. Săculețul i conține m_i monede de aceeași valoare V_i . Elaborați un program care afișează la ecran modul de plată a unei sume p cu monedele din săculețe.

4) Dintr-un fișier se citesc numele și notele medii a x clase de elevii. Fiecare clasă este formată din minim a și maxim b studenți ($2 < a < b < 30$). Studenții necesită a fi grupați în grupuri de câte x studenți, din fiecare grupă este selectat câte un student, astfel încât media notelor grupului să fie egală cu 8. Afișați lista tuturor grupelor, care pot fi formate, enumerând numele elevilor din fiecare grupă.

5) Se dau N ($N \leq 50.000$) dominouri. Să se determine o modalitate de construire al unui șir, care să conțină toate dominourile, respectând regula jocului domino. Aceasta regula prevede ca numerele înscrise pe fețele corespunzătoare a două dominouri consecutive trebuie să fie egale. Dominourile pot fi alese în ordine oarecare și rotite.

Date de intrare. Prima linie a fișierului de intrare *domino.in* se va afla numărul N al dominourilor. Pe următoarele N linii se vor afla câte două numere separate printr-un singur spațiu, reprezentând cele două numere înscrise pe dominoul corespunzător.

Date de ieșire. Pe prima linie a fișierului *domino.out* va trebuie să afișați 1 , dacă există soluție și 0 dacă nu. Dacă există soluție vor urma exact N linii, care descriu șirul construit. Fiecare număr de ordine va fi urmat de un spațiu și de un număr, care poate fi 0 sau 1 și va fi egal cu 1 în cazul în care dominoul corespunzător a fost rotit.

- 1) Utilizând metoda desparte și stăpânește
 - A) Să se determine produsul a n numere întregi;
 - B) Să se determine maximul (minimul) a n numere întregi;
 - C) Să se determine cel mai mare divizor comun a n valori dintr-un vector;
 - D) Să se caute o valoare într-un vector. Dacă se găsește se va afișa poziția pe care s-a găsit, altfel se va afișa mesajul Nu exista asa valoare;
 - E) Să se caute o valoare într-un vector ordonat crescător;
 - F) Să se numere câte valori sunt egale cu x dintr-un șir de numere întregi citite.

2) Considerăm un șir de n numere întregi, ordonat crescător și un număr întreg x . Scrieți un program care împarte șirul dat în două subșiruri în așa fel încât toate elementele primului subșir să fie mai mici sau cel mult egale cu x , iar toate elementele celui de-al doilea subșir sa fie strict mai mari decât x .

3) **Turnurile din Hanoi.** Se dau trei tije, a , b și c . Pe tija a se află n discuri de dimensiuni diferite, ordonate în ordinea diametrelor (discul cel mai mare la bază). Se dorește mutarea tuturor discurilor de pe tija a pe tija b , utilizând tija intermediară c , cu condiția că un disc cu diametrul mai mare să nu fie pus pe vreo tijă peste un disc cu dimensiune mai mică.

4) Intr-o placă dreptunghiulară cu dimensiunile N pe L (coordonatele dreptunghiului) sunt făcute K găuri. Coordonatele găurilor sunt numere întregi. Fișierul *date.in* conține în prima linie numărul n , care reprezintă numărul de găuri, pe liniile următoare se regăsesc coordonatele găurilor. Scrieți un program, care determină cel mai mic dreptunghi(cu o suprafață mai mare ca 0) fără găuri din placă.

5) Cunoscându-se numărul n al elevilor unei clase, precum și mediile generale ale celor n elevi la finele unui an școlar, realizați un program care, folosind un algoritm Divide et Impera, testează dacă în clasa respectivă exista doi elevi cu aceeași medie generală. Programul va tipări pe ecran mesajul “da” sau “nu”, în funcție de situație.

6) Intr-o placă dreptunghiulară cu dimensiunile N pe L sunt n puncte ($2 \leq n \leq 30$). Fișierul *date.in* conține în prima linie numărul n , care reprezintă numărul de puncte din plan pe liniile următoare se regăsesc coordonatele punctelor. Elaborați un program care afișează la ecran coordonatele punctelor P_a , P_b distanța dintre care este minimă. Distanța dintre punctele P_j , P_m se calculează cu ajutorul formulei:

$$d_{jm} = \sqrt{(x_j - x_m)^2 + (y_j - y_m)^2}$$

- 1) Se consideră tabloul bidimensional $T[1:5,1:6]$. Elementele lui pot avea valori 0 sau 1. Determinați numărul „insulițelor” formate de valorile 1.
- 2) Se consideră tabloul bidimensional $T[1:6,1:6]$. Elementele lui sunt numere întregi pozitive. Să ne imaginăm că peste un element curge vopsea care se prelinge peste elementele vecine, valoarea cărora nu o depășește pe a celor peste care s-a scurs vopseaua. Curgerea vopselei are loc pe direcțiile orizontală și verticală. Determinați toate elementele peste care s-a prelins vopseaua.
- 3) Se dau 1000 de monede, dintre care una este mai ușoară decât celelalte, care sunt identice. Determinați moneda falsă printr-un număr minim de cântăriri cu o balanță fără greutate marcate.
- 4) Pe un câmp dreptunghiular se află diferite obiective militare de formă dreptunghiulară. Lățimea lor este egală cu o unitate, iar lungimea poate fi diferită, dar constituie numere întregi de unități de măsură. Determinați numărul de obiective de fiecare tip.
- 5) Un tablou bidimensional are valorile componentelor numere întregi pozitive. Determinați patru elemente ce ar reprezenta vârfurile unui pătrat, astfel încât suma acestor elemente să fie maximală.
- 6) Se dă mulțimea $A=\{1, 2, \dots, n\}$. Să se determine toate permutările acestei mulțimi.
- 7) Se dă mulțimea $A=\{1, 2, \dots, n\}$. Să se determine toate combinațiile acestei mulțimi luate câte k elemente.
- 8) Se dă mulțimea $A=\{1, 2, \dots, n\}$. Să se determine toate partițiile acestei mulțimi.
- 9) Un copil știe doar adunarea cu 1 și înmulțirea cu 2. Ajutați-l, pornind de la 1, să obțină numărul 100 printr-un număr minim de operații respective.
- 10) Se dă un tablou liniar de numere întregi ordonate crescător. Determinați dacă o valoare dată m se regăsește printre componentele tabloului printr-un număr minim de comparații.

11) Se consideră două tablouri liniare de numere întregi sortate crescător. Obțineți în baza lor al treilea tablou astfel încât el să se obțină sortat la completare.

12) Un călător dorește să viziteze n orașe, astfel încât să nu treacă de două ori prin același oraș, revenind în orașul de unde a pornit, marcat cu numărul 1. Cunoscând legăturile existente între orașe, se cere să se tipărească toate drumurile posibile pe care le poate efectua călătorul.

13) Determinați toate modalitățile de a plăti o sumă anumită de bani S cu bancnote având valorile nominalelor A_1, A_2, \dots, A_n . Numărul bancnotelor de fiecare nominal este suficient de mare.

14) Avem la dispoziție pânză de 6 culori. Să se determine toate variantele de drapele tricolore ce se pot proiecta, dacă:

- un drapel are în mijloc galben sau verde;
- cele trei culori sunt distincte.

15) Fie un număr natural N . Determinați toate descompunerile sale în sumă de numere prime.

16) Fie N piese. Se cunoaște timpul necesar pentru prelucrarea fiecărei piese la fiecare dintre două strunguri. Determinați în ce ordine trebuie prelucrate piesele la cele două strunguri, știind că piesa poate fi prelucrată la strungul al doilea numai după ce a fost prelucrată la strungul întâi, astfel încât timpul de lucru al strungurilor să fie cel mai mic posibil.

17) Din fișierul `secv.in` se citesc un număr natural n și o matrice pătratică de dimensiune $n \times n$, conținând litere din alfabet. De la tastatură se citește un cuvânt. Verificați dacă respectivul cuvânt se conține pe liniile sau coloanele matricei, parcurse într-un sens sau în altul (de la stânga spre dreapta sau invers, de sus în jos sau invers). Pe ecran se va afișa numărul liniei și coloanei de unde începe cuvântul și sensul de parcurgere. De ex.,

5 e r e m a h e r e b b m e r e b a m e r a e m r e	Cuvântul: mere	Răspuns: De la linia 1 coloana 4 spre stânga De la linia 1 coloana 4 în jos De la linia 3 coloana 2 spre dreapta
--	-------------------	---

18) Aparatele pentru perforarea tichetelor de călătorie în transportul public folosesc 9 puncte de perforare, dispuse în formă de matrice 3×3 . Să se genereze toate modalitățile în care pot fi perforate biletele.

19) Se consideră planul unui labirint, redat prin matricea $[A_{1:n}, 1:m]$, elementele căreia au valori 1, dacă poziția respectivă reprezintă obstacol, sau 0 – în caz contrar ($1 \leq n, m \leq 30$). Pe plan sunt date două puncte B și C, ce nu reprezintă obstacole. Elaborați un program ce determină dacă există un drum din punctul B până în punctul C. Deplasarea se face doar pe direcție orizontală sau verticală.

20) Se dau n șiruri de caractere formate din cifre. Îmbinați aceste șiruri astfel, încât numărul obținut să fie cel mai mare posibil ca valoare.

21) Se consideră două polinoame de mai multe necunoscute. Determinați suma acestor polinoame. De ex., $P_1: 3ax^2 - 2xy + 5ay + 3$; $P_2: 2ax^2 - 3x^2y + 6ay + 2xy$. Răspuns: $5ax^2 + 11ay + 3 - 3x^2y$.

22) Între n orașe există o rețea de drumuri ce permite ca dintr-un oraș să se ajungă în oricare alt oraș. Între două orașe există cel puțin un drum direct de lungime cunoscută. Determinați traseul cu distanța minimă posibilă ce se va parcurge la deplasarea din orașul A_i în orașul A_j .

23) Un șir de caractere conține reprezentarea unui număr scris în sistemul roman de numerație. Determinați echivalentul acestui număr în sistemul zecimal de numerație.

24) Se consideră trei numere naturale A, B, C . Determinați toate numerele naturale mai mici ca N ce pot fi reprezentate sub forma unei sume cu un număr arbitrar de termeni, fiecare dintre termeni fiind A sau B .

25) Numerele a, b, c se numesc pitagoriene dacă ele pot fi lungimile laturilor unui triunghi dreptunghic. Determinați toate numerele pitagoriene mai mici decât o valoare dată n .

26) Fie n localități. Între fiecare două dintre aceste localități există sau nu comunicații aeriene. Fiind date două localități, să se determine dacă se poate ajunge dintr-o localitate în alta pe cale aeriană.

27) Fie o scară cu n (n - număr natural) trepte. O persoană trebuie să urce scara. La fiecare pas ea poate urca una sau două trepte consecutive. Determinați numărul de moduri în care persoana dată poate urca scara.

28) Se dă numărul natural n . Determinați primele n numere naturale, a căror descompunere în factori primi conține doar factori din mulțimea $\{2, 3, 5\}$.

29) Se consideră un dicționar explicativ, păstrat într-u fișier text. Fiind dat un termen nou, includeți termenul și definiția lui în dicționar printr-un număr minim de operații.

30) Președintele unei țări este ales de către parlament din care fac parte n deputați. Pentru a fi ales președintele trebuie să primească cel puțin $2/3$ din voturile deputaților. Între anumiți deputați există conflicte de interese. Doi deputați aflați în conflict de interese votează diferit. Fiind date numărul natural n și perechile de numere x, y , unde deputații cu numărul de ordine x și y au conflict de interese, să se verifice dacă este posibilă alegerea președintelui.

31) Să se scrie toate cifrele de la 1 la 9 în rând, astfel încât fiecare două vecine să formeze un număr divizibil cu 7 sau cu 13.

32) Se dau două tablouri liniare cu N ($N \leq 500$) componente numere naturale cu valori până la 1000. Determinați valorile ce se întâlnesc în ambele tablouri. În tablouri pot fi valori ce se repetă.

33) Scrieți un program ce determină toate numerele naturale pseudoperfecte mai mici decât o valoare dată N . Un număr natural se numește pseudoperfect dacă este un divizor al sumei divizorilor săi.

34) Se consideră 9 pătrate cu laturile 2, 5, 7, 9, 16, 25, 28, 33, 36 unități. Asamblați aceste pătrate astfel încât să se obțină un dreptunghi având laturile de 61 și 69 unități.

35) Se dă o matrice pătratică. Ordonăți crescător elementele fiecărei linii, apoi rearanjați liniile matricei, astfel încât suma elementelor de pe diagonala principală să fie minimă.

36) Se dau M tablouri liniare cu numărul de componente diferit. Numărul de componente ale fiecărui tablou se conține în tabloul $L[1:m]$. Să se determine secvențele maxime de elemente alăturate ce figurează în toate tablourile în aceeași ordine.

37) Într-o localitate sunt 15 vânători. Dintre ei se formează echipe a câte 3 persoane, astfel încât în cele 7 zile cât durează vânătoarea nici un participant să nu fie împreună cu vreunul dintre cei cu care a mai fost în echipă. Determinați componența echipelor pentru cele 7 zile.

38) Se cunosc coordonatele orașelor unei țări. Determinați configurația unei rețele telefonice prin cablu astfel încât: a) oricare oraș să fie conectat la rețeaua telefonică; b) costul rețelei telefonice să fie minim.

39) La o întrunire trebuie să vorbească 5 persoane (A, B, C, D, E). Afișați toate listele de ieșire la tribună astfel încât de fiecare dată persoana B să vorbească după persoana A.

40) La o discotecă sunt M băieți și N fete. Determinați toate posibilitățile în care băieții pot invita fetele astfel încât doi băieți să nu invite aceeași fată și fiecare băiat să danseze ($N \geq M$).

41) Pe o hartă de contur sunt reprezentate n ($n \leq 30$) țări. În memoria calculatorului harta este redată prin matricea $[A]_{1:n, 1:m}$, elementele căreia au valori 1, dacă țările respective sunt vecine, sau 0 – în caz contrar. Determinați numărul minim de culori necesare pentru a colora harta astfel ca oricare două țări vecine să aibă culori diferite. (Utilizând cel mult patru culori, colorați harta astfel încât două țări cu frontieră comună să fie colorate diferit.)

Capitolul 5

PROBLEME CU CARACTER DE OLIMPIADĂ

Vopsit. Doi copii vopesc un gard din scânduri pe care le vom numerota de la 1 la n astfel: primul ia o cutie de vopsea roșie cu care vopsește scândurile cu numărul p , $2p$, $3p$, etc. Al doilea procedează la fel, începe de la același capăt al gardului dar ia o cutie de vopsea albastră și vopsește din q în q scânduri. Astfel, când vor termina de vopsit, gardul va avea multe scânduri nevopsite, unele scânduri vopsite în roșu, altele în albastru, iar altele în violet. Cunoscând numerele n , p și q afișați: a) câte scânduri rămân nevopsite b) câte scânduri sunt vopsite în roșu c) câte scânduri sunt vopsite în albastru d) câte scânduri sunt vopsite în violet.

Paranteze. Se consideră o expresie aritmetică, unde sunt folosite parantezele $(, \{, [,], \}$ și $)$. Elaborați un program care verifică dacă în expresie parantezele au fost folosite corect.

Input: Fișierul de intrare *paranteze.in* care conține pe fiecare linie câte o expresie matematică.

Output Fișierul de ieșire *paranteze.out* va conține pe fiecare linie cuvântul DA sau NU în funcție de corectitudinea șirului respectiv din fișierul *paranteze.txt*.

Exemplu:

Fișierul input.txt	Fișierul output.txt:
$(a*b)$	DA
$(12-e)(d+a)[c($	NU

Film. F fete și B băieți și-au luat bilete la film. Toate biletele sunt pe același rând, pe scaune consecutive. Elaborați un program care să determine o posibilitate de aranjare a fetelor și băieților pe scaune astfel încât lângă orice băiat să se afle cel puțin o fată, iar lângă orice fată să stea cel puțin un băiat.

Input: Fișierul de intrare *film.in* conține pe prima linie două numere naturale separate prin spațiu $F B$, unde F reprezintă numărul de fete, iar B numărul de băieți.

Output: Fișierul de ieșire *film.out* va conține o singură linie pe care vor fi scrise F litere 'f' și B litere 'b' ('f' indicând o fată, iar 'b' indicând un băiat), respectând condițiile problemei.

Restricții: $1 \leq F, B \leq 100$

Pentru datele de test exista întotdeauna soluție.

Exemplu:

<i>film.in</i>	<i>film.out</i>
3 4	bfbfbfb

Finala. La finala campion 2009 vor participa N elevi și M profesori. Organizatorii colaborează cu o firmă de transport care are autobuze identice, fiecare autobuz având K locuri. Elaborati un program ce va determina numărul minim de autobuze necesare, știind că în orice autobuz trebuie să existe cel puțin 2 profesori.

Input: Fișierul de intrare *finala.in* conține pe prima linie trei numere naturale separate prin câte un spațiu $N M K$, cu semnificația din enunț.

Output: Fișierul de ieșire *finala.out* va conține un singur număr natural reprezentând numărul minim de autobuze necesare. Dacă nu este posibilă organizarea transportului în condițiile problemei, veți afișa valoarea 0.

Restricții: $1 \leq N, M, K \leq 10000$

Exemplu:

<i>finala.in</i>	<i>finala.out</i>	Explicații
10 4 7	2	Sunt 10 copii și 4 profesori. În fiecare autobuz vor merge 5 copii și 2 profesori, deci sunt necesare 2 autobuze
10 4 5	0	Transportul nu poate fi organizat în condițiile date

Petrolul. Republica Moldova, trecută prin criza gazelor, și neavând resurse de petrol și-a propus să procure petrol de peste hotare.

Există n țări care exportă petrol. Fiecare i -stat exportă petrol cu a_i dolari și b_i euro pentru un baril. Moldova nu poate procura petrol de la un stat și în dolari și în euro. Ajutați Republica Moldova să procure petrol cât mai mult, dacă se cunoaște numărul a (numărul de dolari de care dispune RM) și b (numărul de euro de care dispune RM).

Input: Fișierul de intrare *oil.in* conține în primul rând 3 numere: n, a, b ($1 \leq n \leq 100, 0 \leq a, b \leq 1000$). Următoarele N rânduri conțin perechile de numere a_i și b_i

Output: Fișierul de ieșire *oil.out* va conține volumul maxim de petrol care va putea fi procurat cu minim 2 cifre după virgulă.

Exemplu:

oil.in	oil.out
3 2 5 6 4 3 5 8 7	1.92
4 3 2 1 1 2 2 3 3 4 4	4.00

Evaluarea expresiilor Se consideră expresiile aritmetice formate din numere întregi, mai mici decât 1000 și operatorii binari +, −, *. Scrieți un program care evaluează expresiile aritmetice în studiu.

Input: Fișierul *input.txt* va conține expresia de evaluat.

Output: Fișierul *output.txt* va conține rezultatul evaluării.

Restricții: Lungimea maximă a expresiei este de 300 de caractere.

Exemplu:

Fișierul input.txt	Fișierul output.txt:
16+34+2*4-7	51

Parcurgerea matricei. Se dă matricea $A[N,N]$ de numere întregi ($N \leq 10$). De parcurs matricea în următoarea ordine: primul se va afișa elementul $A[N,1]$, urmează mișcarea în dreapta, apoi paralel cu diagonala principală, mișcarea în sus, apoi iarăși paralel cu diagonala principală, ș.a.m.d., ultimul element va fi $A[1,N]$.

Input: De la tastatură se introduce N și elementele matricei

Output: La ecran se vor afișa elementele în ordinea indicată

Exemplu: Input: 4

```

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

Output:

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

Codificare optimală Se dă un text de lungime maximă 100 caractere, ce conține doar litere mici ale alfabetului englez. Textul poate fi codificat înlocuind aparițiile consecutive ale subșirurilor sale de lungime maximă cu subșirul urmat de numărul său de apariții. Scrieți un program care să codifice optimal un text dat, adică codul rezultat să fie de lungime minimă.

Input: fișierul *cod.in* conține pe o linie textul.

Output: fișierul *cod.out* va conține codul.

Exemplu:

Fișierul cod.in	Fișierul cod.out
aaacaaacaaacbbdefdef	aaac3b2def2

Șirul Fibonacci[12] Șirul de numere Fibonacci este definit în felul următor: primele două elemente ale șirului sunt egale cu 1, fiecare element următor este suma celor două elemente care îl precedă. Șirul de numere Fibonacci este scris fără spații, astfel încât începutul lui arată în felul următor:

11235813213455...

Cerință: Se cere să scrieți un program care să determine cifra de pe poziția N din șirul obținut (pozițiile în șir sunt numerotate începând cu 1).

Date de ieșire: va fi afișat un număr natural – valoarea cifrei de pe poziția N.

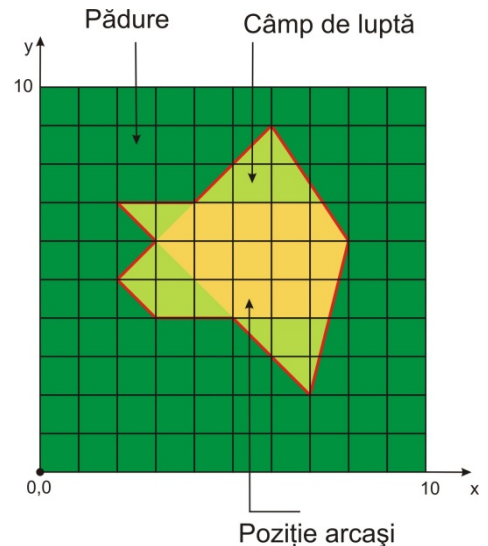
Restricții

$$1 \leq N \leq 10000000$$

Exemple:

Date de intrare	Date de ieșire	Explicații
6	8	112358
Date de intrare	Date de ieșire	Explicații
14	5	11235813213455

Arcași[12] Secretul victoriilor faimosului comandant de oști MegaFlop este strategia lui de alegere a poziției arcașilor pe câmpul de luptă. Câmpul de luptă are forma unui poligon simplu și e înconjurat de păduri. MegaFlop plasează arcașii doar pe poziții din care este **văzut** tot câmpul de luptă. Se consideră că arcașii văd tot câmpul, dacă din orice punct care aparține poziției lor de tragere se poate trage cu săgeata în orice alt punct al câmpului. Traectoria săgeții este liniară. Nimerind în pădure, săgeata se pierde. Pentru tragere, fiecare arcaș are nevoie de 1 unitate de suprafață. Astfel, numărul maxim de arcași, care pot fi plasați pe poziții este determinat de aria poligonului din care este văzută toată câmpia. (des. 1)



Cerință. Scrieți un program, care determină numărul maxim de arcași, care pot fi plasați pe poziții pe câmpul de luptă.

Input. Fișierul de intrare va conține pe prima linie un număr întreg N - numărul de vârfuri ale poligonului simplu, care descrie perimetrul câmpului de luptă. Urmează N linii care conțin coordonatele vârfurilor poligonului în ordinea parcurgerii lor după acele de ceasornic, câte un vârf pe linie. Linia $i+1$ conține două numere întregi x_i, y_i , separate prin spațiu – coordonatele vârfului i .

Output. Fișierul de ieșire va conține un singur număr întreg: numărul maxim de arcași, care pot fi plasați pe poziții.

Restricții

$$3 \leq N \leq 1\,000, \quad 0 < x_i, y_i \leq 10\,000$$

Exemplu

arcas.in (desen)	arcas.out
8 2 5 3 6 2 7 4 7 6 9 8 6 7 2 5 4 3 4	11
arcas.in	arcas.out
5 1 3 2 6 2 3 5 6 3 1	0

Cetăți[12] Terralanda este un regat, care se extinde permanent. Frontiera regatului este un poligon convex, în vârfurile căruia sunt construite cetăți pentru apărarea segmentelor de frontieră adiacente. La fiecare extindere a regatului frontiera se modifică: se construiesc cetăți noi, iar unele cetăți, construite anterior își pierd rolul de puncte de apărare. În toate cetățile se află garnizoane. Pentru a micșora cheltuielile pentru întreținerea armatei, regele Terralandei, MegaBit a decis să demobilizeze garnizoanele din cetățile care nu mai au rolul de puncte de apărare (nu se mai află pe frontiera regatului).

Cerință. Scrieți un program, care va determina din câte cetăți vor demobilizate garnizoanele.

Date de intrare. Prima linie a fișierului de intrare cetati.in conține un număr întreg: N ($3 \leq N \leq 10000$) – numărul de cetăți în Terralanda.

Urmează N linii ce conțin câte două numere întregi x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), separate prin spațiu – coordonatele cetăților.

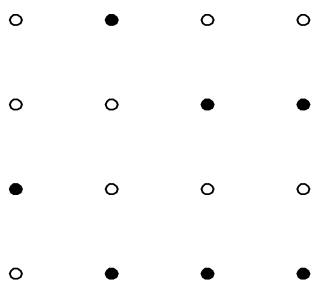
Date de ieșire. Fișierul de ieșire cetati.out va conține un număr întreg – numărul de cetăți din care pot fi demobilizate garnizoanele.

Exemplu:

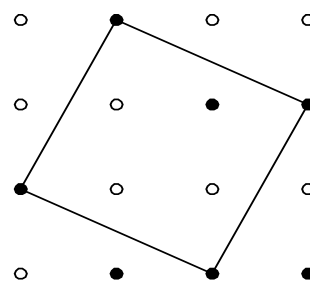
cetati.in	cetati.out	Explicație
10 2 1 3 4 6 3 6 6 8 5 10 3 11 7 12 6 9 11 15 8	5	

Pătrate[12] Fie dată o rețea din puncte cu coordonate întregi $N \times N$. Fie unele noduri ale rețelei sunt colorate cu alb, altele – cu negru. Se cere să determinați numărul de pătrate pe rețeaua dată (vârfulurile unui pătrat trebuie să coincidă cu nodurile rețelei și să fie colorate cu aceeași culoare).

De exemplu, pentru rețeaua 4×4 , din desenul 1 există un singur pătrat (des. 2).



Des 1. Rețea 4×4 .



Des 2. Pătratul din rețea.

Input. Prima linie a fișierului de intrare conține numărul N – dimensiunea rețelei ($2 \leq N \leq 50$). Următoarele N linii conțin câte N numere din mulțimea $\{0, 1\}$ și descriu rețeaua. Dacă nodul cu coordonatele (i, j) este de culoare albă, atunci elementul j din linia $i+1$ este 0, iar dacă e de culoare neagră - atunci 1.

Output. Fișierul de ieșire va conține un singur număr – cel al pătratelor din rețea.

Exemplu

patrate.in	patrate.out
4 0 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1	1

PROTOZUMA[12] ProtoZuma este un joc foarte simplu, în care un șir S de simboluri (simbolurile sunt numerotate de la stânga la dreapta, începând cu indicele 1) este lovit în poziția **i** de o ghiulea. Imediat după lovitură se produce „explozia”, după următoarea regulă:

- Dacă simbolurile din pozițiile **i-1, i, i+1** nu sunt egale între ele, nu se întâmplă nimic – șirul S rămâne intact.

Exemplu: **AAABB** este lovit în poziția 4. *Rezultat:* **AAABB**

- Dacă simbolurile din pozițiile **i-1, i, i+1** sunt egale, din șir se lichidează toată secvența de simboluri egale, care le conține, iar fragmentele rămase se unesc.

Exemplu: **BBAAAACCC** este lovit în poziția 4 *Rezultat:* **BBCCC**

- Atât timp cât după explozie în locul unirii fragmentelor se formează o secvență de cel puțin 3 litere egale, explozia se repetă.

Exemplu: **DBBAAAABCCC** este lovit în poziția 6

Rezultat: **DBB <-> BCCC**, explozia se repetă, *rezultat final* **DCCC**

Cerință: Scrieți un program, care să determine șirul în care se transformă șirul inițial S după lovitură a unei ghiulele.

Restricții: Lungimea S nu va depăși 200 caractere. $1 < i < \text{lungimea șirului S}$

Input: Fișierul de intrare **x-zuma.in** va conține pe prima linie un număr natural – valoarea **i** a poziției în care lovește ghiuleaua. Linia a doua a fișierului conține șirul S

Output: Fișierul de ieșire **x-zuma.out** va conține pe prima (și unica) linie șirul S după lovitură și explozii.

Exemplu:

zum.in	zum.out
12 AABBCCCCBRQQQAA	AABBCCCCBRAA
6 AABBCCCCBRQQQAA	AARQQQAA
5 AABBCCCCBRQQQAA	AABBCCCCBRQQQAA

Mere. În grădina lui Ion crește un pom de măr. Pomul fiind fermecat, merele cad din el după o regulă diferită de legile fizicii: La un moment dat un măr de desprinde de ramura sa și începe să cadă vertical în jos. Dacă în timpul căderii mărul atinge un alt măr, acesta din urmă începe să cadă și el vertical în jos. Traectoria merelor în cădere nu se schimbă la atingerea altor mere. Astfel, orice măr, cu excepția primului, începe să cadă doar după ce a fost atins de un măr în cădere.

Cerință Se cere să scrieți un program, care să determine numărul de mere, care vor cădea din pom.

Date de intrare. Fișierul de intrare **mere.in** va conține pe prima linie numărul **n** al

merelor din pom. Următoarele n linii conțin descrierea merelor. Linia $i+1$ conține descrierea mărului cu indicele i . Fiecare măr este considerat o sferă. Un măr este descris de coordonatele punctului său superior (în acest punct el este prins de ramură, codița fiind punctiformă) x_i, y_i și z_i și raza r_i toate numerele - întregi). Se garantează, că inițial nici care două mere nu se intersectează. Axa OZ este orientată vertical în sus.

Date de ieșire. Fișierul de ieșire mere.out va conține un singur număr - cel al merelor, care vor cade din copac în cazul când începe să cadă mărul cu indicele 1 .

Restricții

$$1 \leq N \leq 200$$

$$-10000 \leq (x_i, y_i, z_i) \leq 10000$$

$$1 \leq r_i \leq 10000$$

Exemplu

mere.in	mere.out	Explicații
4 0 0 10 4 5 0 3 1 -7 4 7 1 0 1 2 6	3	Mărul cu indicele 1 va cădea în orice caz. În cădere el atinge mărul cu indicele 2 și mărul cu indicele 4.

Capitolul 6

INFORMAȚII UTILE ÎN TIMPUL LUCRULUI ÎN MEDIUL TURBO PASCAL

CODURILE GENERATE DE TASTATURĂ

Tasta	Normal	Shift	Ctrl
A	97	65	1
B	98	66	2
C	99	67	3
D	100	68	4
E	101	69	5
F	102	70	6
G	103	71	7
H	104	72	8
I	105	73	9
J	106	74	10
K	107	75	11
L	108	76	12
M	109	77	13
N	110	78	14
O	111	79	15
P	112	80	16
Q	113	81	17
R	114	82	18
S	115	83	19
T	116	84	20
U	117	85	21
V	118	86	22
W	119	87	23
X	120	88	24
Y	121	89	25
Z	122	90	26
0)	48	41	
1!	49	33	
<u>2@</u>	50	64	
3#	51	35	
4\$	52	36	
5%	53	37	
6^	54	94	
7&	55	38	
8*	56	42	
9(57	40	

Tasta	Normal	Shift	Ctrl
~	96	126	
-_	45	95	
=+	61	43	
[{	91	123	
]}	93	125	
\	92	124	
::	59	58	
' "	39	34	
,<	44	60	
.>	46	62	
/?	47	63	
Backspace	8	8	127
Tab	9	0,15	
Enter	13	13	10
Spațiu	32	32	32
Esc	27	27	27
F1	0,59	0,84	0,94
F2	0,60	0,85	0,95
F3	0,61	0,86	0,96
F4	0,62	0,87	0,97
F5	0,63	0,88	0,98
F6	0,64	0,89	0,99
F7	0,65	0,90	0,100
F8	0,66	0,91	0,101
F9	0,67	0,92	0,102
F10	0,68	0,93	0,103
PrintScr			0,114
Insert	0,82	0,5	0,4
Home	0,71	0,71	0,119
PageUp	0,73	0,73	0,132
Delete	0,83	0,7	0,6
End	0,79	0,79	0,117
PageDn	0,81	0,81	0,118

Valorile pentru litere se refera la situația când <Caps Lock> este inactiva. Când <Caps Lock> este activata, valorile coloanelor *Normal* si *Shift* se inversează. Tastele blocului numeric transfera, la tastarea normala, codurile pentru cifre (#48..#57), numai daca <Num Lock> este activ. În toate celelalte cazuri, se transferă codurile funcțiilor asociate, precizate în tabelul anterior. Tastele **F11**, **F12**, **Scroll**, **Pause** nu generează coduri.

Erori de execuție

Apariția unei erori de execuție determina întreruperea programului si afișarea unui mesaj de eroare, de forma: **Run-time error nnn at xxxx:yyyy**, unde nnn este codul erorii de execuție, iar xxxx:yyyy este adresa ei (**segment si offset**). Erorile de execuție se împart în: erori DOS (coduri 1-99); erori de intrare/ieșire (coduri 100-149), erori critice (coduri 150-199) si erori fatale (coduri 200-255).

ERORI DOS

1 Funcție inexistentă. Generata de un apel al unei funcții DOS inexistente.

2 Fișier inexistent. Generata de execuția uneia din procedurile *Reset*, *Append*, *Rename* sau *Erase*, daca identificatorul asignat variabilei de tip fișier nu corespunde unui fișier existent.

3 Cale inexistentă. Generata de execuția uneia din procedurile:

- o *Reset*, *Append*, *Rewrite*, *Rename* sau *Erase*, daca identificatorul asignat variabilei de tip fișier este invalid sau include un sub[director] inexistent;

- o *ChDir*, *MkDir* sau *Rmdir*, daca sub[directorul] este invalid sau inexistent.

4 Prea multe fișiere deschise. Generata de execuția uneia din procedurile *Reset* sau *Append* daca, la un moment dat, sunt deschise simultan mai mult de 12 fișiere ale utilizatorului. Daca se dorește raportarea erorii pentru un numar mai mic de fișiere deschise simultan, trebuie ca fișierul **CONFIG.SYS** sa nu con-tina clauza **FILES=xx**, sau sa specifice numărul de fișiere dorit.

5 Acces interzis la fișier. Generata de execuția uneia din procedurile:

- o *Reset* sau *Append*, daca *FileMode* permite scrierea, dar identificatorul asignat variabilei fișier specifica un [sub]director/fișier read-only;

- o *Rewrite*, daca sub[directorul] este plin sau identificatorul asignat variabilei fișier specifica un [sub]director/fișier existent read-only;

- o *Rename*, daca identificatorul asignat variabilei fișier specifica un fișier existent;

- o *Erase*, daca identificatorul asignat variabilei fișier specifica un sub[director]/fișier read-only;

- o *Mkdir*, daca: exista un fișier cu aceleași nume în sub[directoriu] părinte; nu exista spațiu în sub[directorul] părinte; este specificat în cale un dispozitiv;

- o *Rmdir*, daca: sub[directorul] nu este vid; nu se specifica un sub[director] în cale; directorul specificat include rădăcina;

- o *Read/BlockRead* pentru un fișier cu tip/fără tip, daca acesta nu a fost deschis pentru citire;

- o *Write/BlockWrite* pentru un fișier cu tip/fără tip, daca acesta nu a fost deschis pentru scriere.

6 Handle de fișier invalid. Generata la transmiterea unui handle (vezi §8.2) invalid de fișier, la un apel al sistemului DOS.

- 12 Cod invalid de acces la fișier.** Generata de execuția uneia din procedurile *Reset* sau *Append* pentru fișiere cu tip/fără tip, daca valoarea variabilei *FileMode* este invalida.
- 15 Număr dispozitiv invalid.** Generata de execuția uneia din procedurile *GetDir* sau *ChDir*, daca numărul dispozitivului periferic este invalid.
- 16 Sub[directorul] curent nu poate fi suprimat.** Generata de execuția procedurii *Rmdir*, daca în calea specificata este inclus directorul curent.
- 17 Redenumire fișiere pe dispozitive diferite.** Generata de execuția procedurii *Rename*, daca specificatorii de fișiere nu sunt pe același dispozitiv.

ERORI DE INTRARE/IESIRE

Erorile de intrare/ieșire determina întreruperea execuției programului, numai daca instrucțiunea respectiva a fost compilata cu directiva $\{ \$I+ \}$ (valoare implicită). În cazul în care se specifica directiva de compilare $\{ \$I- \}$, execuția programului continuă, iar apariția erorii este depistata cu ajutorul funcției *IOResult*.

- 100 Eroare la citirea de pe disc.** Generata de execuția procedurii *Read* pentru fișiere cu tip, daca se încearcă citirea sfârșitului de fișier.
- 101 Eroare la scrierea pe disc.** Generata de execuția uneia din procedurile *Close*, *Write*, *WriteLn*, *Flush* sau *Page*, daca s-a umplut discul (nu mai este spațiu pe disc).
- 102 Fișier neassignat.** Generata de execuția uneia din procedurile *Reset*, *Rewrite*, *Append*, *Rename* sau *Erase*, daca variabila fișier nu a fost asignata unui nume fizic, prin procedura *Assign*.
- 103 Fișier nedeschis.** Generata de execuția uneia din procedurile/funcțiile *Close*, *Read*, *Write*, *Seek*, *Eof*, *FilePos*, *FileSize*, *Flush*, *BlockRead* sau *BlockWrite*, daca fișierul nu este deschis.
- 104 Fișier nedeschis pentru intrare.** Generata de execuția uneia din procedurile/funcțiile *Read*, *ReadLn*, *Eof*, *EoLn*, *SeeKEof* sau *SeeKEoLn*, daca fișierul **TEXT** respectiv nu este deschis pentru consultare.
- 105 Fișier nedeschis pentru ieșire.** Generata de execuția uneia din procedurile *Write* sau *WriteLn*, daca fișierul **TEXT** respectiv nu este deschis pentru crea-re/ex-tindere.
- 106 Format numeric invalid.** Generata de execuția uneia din procedurile *Read* sau *ReadLn*, daca o valoare numerica citita dintr-un fișier **TEXT** nu concorda cu formatul numeric declarat.

ERORI CRITICE

- 150 Disc protejat la scriere**
- 151 Unit necunoscut**
- 152 Dispozitivul nu este pregătit**
- 153 Comanda necunoscuta**
- 154 Eroare CRC în data**
- 155 Cerere pe un dispozitiv greșit**
- 156 Eroare de poziționare pe disc**
- 157 Tip dispozitiv necunoscut**
- 158 Sector negăsit**
- 159 Imprimanta în așteptarea hârtiei**

160 Incident la scrierea pe dispozitiv

161 Incident la citirea de pe dispozitiv

162 Întrerupere hardware

ERORI FATALE

200 Împărțire la zero. Generata de împărțirea la 0 a unui numar, cu operatorii / , MOD sau DIV.

201 Nonapartenența la un interval. Generata de instrucțiunile compilate cu directiva {\$R+}, în următoarele condiții:

- o expresia de indice pentru referirea unui element de masiv este în afara intervalului;
- o atribuirea unei valori în afara intervalului stabilit pentru variabila respectiva;
- o atribuirea unei valori în afara intervalului stabilit pentru un parametru de

procedura/funcție.

202 Depășire stiva. Generata la apelul unei proceduri/funții, compilate cu directiva {\$S+}, când nu este spațiu suficient în stiva pentru memorarea variabilelor locale. Stiva se poate mari cu directiva de compilare {\$M}. Eroarea apare si în cazul unui apel recursiv infinit.

203 Depășire heap. Generata de execuția uneia din procedurile New sau *GetMem*, când nu este suficient spațiu în heap, pentru alocarea unui bloc sau a unei zone de mărime specificata.

204 Operație cu pointer invalid. Generata de execuția uneia din procedurile *Dispose* sau *FreeMem* daca: pointerul are valoarea nil sau indica o locație în afara zonei heap; lista blocurilor libere nu poate fi extinsa, deoarece este plina; *HeapPtr* are o valoare prea apropiata de limita inferioara a listei libere.

205 Depășire virgula mobila. Generata în urma unei operații al cărei rezultat este un numar prea mare pentru a fi reprezentat într-un tip real de data **Pascal**.

206 Depășire inferioara virgula mobila. Generata în urma unei operații al cărei rezultat este un numar prea mic pentru a fi reprezentat într-un tip real de data **Pascal**. Apare numai daca se utilizează coprocesorul matematic 8087. Se transmite, implicit, valoarea zero.

207 Operație virgula mobila invalida. Generata daca:

- o Argumentul funcțiilor *Trunc* sau *Round* este în afara intervalului [-2147483648, 2147483647];
- o Argumentul funcției *Sqrt* este negativ;
- o Argumentul funcției *Ln* este negativ sau zero;
- o A apărut o depășire a stivei 8087.

208 Managerul de reacoperire nu este instalat. Generata în urma apelului unei funcții/proceduri de reacoperire, în cazul în care componenta de gestiune a structurilor de reacoperire (*Overlay Manager*) nu a fost instalata (cel mai adesea nu s-a apelat procedura *OvrInit* sau apelul ei a eșuat).

209 Eroare la citirea unui fișier de reacoperire. Generata în cazul în care se produce o eroare când managerul de reacoperire încearcă sa citească un unit dintr-un fișier de reacoperire.

BIBLIOGRAFIE

1. Bacalaureat 2002 teste la informatică, G. Vasilache, I. Ciobanu, A. Malearovici, I. Spinei, L. Țurcanu, Liceum 2002;
2. Cornelia Ivașc, Mona Prună, Tehnici de programare (aplicații de laborator) Ed. Petrion, București, 2000
3. Culegere de problem de informatică, Gh. Bostan, Editura Lumina, Chișinău, 1996;
4. Florin Munteanu, Traian Ionescu, Daniela Tătaru, Sergiu Mihai Dascală, Gheorghe Muscă., Programarea calculatoarelor, manual pentru licee de informatică clasele X-XII, Ed. Didactică și Pedagogică, R.A, - București, 1994
5. Ilie Coandă, Pascal programe-exemple LITERA, Chișină -1998
6. Informatica limbajul PASCAL, manual pentru clasele 9 – 11, A. Gremalschi, I. Mocanu, I. Spinei, Chișinău, Î.E.P. Știința, 2005;
7. Ion Ivan, Marian Dâradală, Felix Furtună, Informatica, Manual pentru clasa a X-a, București, Corint, 2000
8. Ion Smeureanu, Ion Ivan, Mmarian Dâradală., Limbajul C/C++ prin exemple, Ed. CISON, București, 1996
9. Tudor Sorin, Informatica(Tehnici de programare), Varianta C++, manual pentru clasa a X, Ed. L&S, București
10. Tudor Sorin, Informatica, Varianta C++, manual pentru clasa a IX, Ed. L&S, București
11. Tudor Sorin, Informatica, Varianta C++, manual pentru clasa a XI, Ed. L&S, București
12. <http://campion.edu.ro/>