

**REVISTA SCOLII GIMNAZIALE BALCESTI**

**Nr. 5 2021**



**COLECTIVUL DE REDACŢIE**

**COORDONARE:**

**FONDATOR BOGDAN CONSTANTIN**

**REDACTOR- ŞEF: DOBRE ROXANA , BUJOR MARIA MIHAELA**

**CONSULTANT: COJOCARU MIHAELA , CEPOI DELIA , RADOI CARMEN**

**TEHNOREDACTARE:OPREA RADU, IENCUT CRISTINA**

**CORECTOR:COJAN GEORGIANA**

**PUBLICARE REVISTA:BOGDAN CONSTANTIN**

**Adresa redactiei:**

BALCESTI , COMUNA BENGESTI CIOCADIA

**Fiecare autor îşi asumă responsabilitatea pentru conţinutul textului publicat.**

**APLICATII INFORMATICE PENTRU LABORATOR**

# PROF DR BOGDAN CONSTANTIN

**CAPITOLUL I**

**INSTRUCTIUNI**

|  |  |
| --- | --- |
| **§1** | **Operaţii de intrare/ieşire** |

|  |  |
| --- | --- |
| *Operaţii de ieşire* | |
| **Pascal** | write(x); sau writeln(x);  write(x1,x2,…xn); este echivalent cu write(x1); write(x2,); …write(xn);  Parametrii din apel pot avea una din formele:  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.  Fie următoarea secvență de program:  Var a:integer; b:real; ... a:=5; b:=7.236; writeln(’Valoarea lui a este’,a:6, ’ b=’,b:5:2)  va afișa:  Valoarea lui a este 5 b= 7.24  Atragem atenția că ultima cifră este 4 și nu 3 fiindcă a avut loc rotunjirea.  Instrucțiunea *writeln* se deosebește de *write* prin faptul că după afișarea tuturor parametrilor se trece în linie nouă. |
| **C** | printf(sir\_de\_ieşire,listă\_de\_ieşire);  unde :  *şir\_de\_ieşire* este un şir în componenţa căruia intră caractere ordinare şi grupe de caractere speciale, ce se numesc formate sau specificaţii de conversie.  *listă\_de\_ieşire* reprezintă enumerarea elementelor de ieşire, ce se despart prin virgulă. În calitate de elemente pot fi constante, variabile şi expresii.  *Exemplu* :  Fie date următoarele declaraţii :  **int** x=4,y=6;  printf("%d+%d=%d”,x,y,x+y);  În rezultatul execuţiei acestei secvenţe de program va fi afişat mesajul  4+6 =10.  **int** a=24;  **double** c=123.1234567;  instrucţiunea  printf(“%d\n a=%5d\nc=%7.2g”,a,a,c);  va afişa :  24  a= 24 numărul a este afişat pe 5 poziţii  c= 123.12 numărul c este afişat pe 7 poziţii cu 2 cifre după virgulă |
| **C++** | cout<<flux1<<flux2<<…<<fluxn;  Fie date următoarele declaraţii :  **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;  va afişa :  24  a= 24 //numărul a este afişat pe 5 poziţii  c= 123.12 // numărul c este afişat pe 7 poziţii cu 2 cifre după virgulă  xxxxxxxx77 |
| *Operaţii de intrare* | |
| **Pascal** | read(x); sau readln(x);  read(x1,x2,…xn); este echivalent cu  read(x1); read(x2,); …read(xn);  La execuția acestei instrucțiuni compilatorul așteaptă de la tastatură valoare (valori) pentru parametrul (parametrii) înscriși 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.  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.  Fie următoarea secvență de program:  Var a,b:integer;;  Begin write (’Introduceti doua numere intregi ’) readln(a,b);…  sau  Begin write (’Introduceti valoarea lui a ’) readln(a);  write (’b=’) readln(b); … |
| **C** | Pentru introducerea datelor cu format se foloseşte funcţia *scanf*(), este definită în biblioteca *stdio*.h. Mod de utilizare :  scanf(şir\_de\_control,lista\_de\_intrare);  unde :  *şir\_de\_control* este un şir scris în ghilimele. În cadrul acestui şir se indică tipul valorilor care urmează a fi citite. Şirul este format din % şi caracterul de conversie (sunt date mai sus).  *lista\_de\_intrare* indică adresele variabilelor pentru care vor fi citite valori. În faţa fiecărei variabile va fi operatorul adresă &. În cazul în care se citesc mai multe valori acestea vor fi despărţite prin virgulă.  *Exemplu*  Fie date următoarele declaraţii :  **int** a;  **float** b;  Pentru citirea valorilor de la tastatură a acestor variabile într-o singură instrucţiune vom scrie :  scanf(”%d%f”,&a,&b);  *%d* indică tipul variabilei a (*int*), pe când *%f* indică tipul variabilei b (*float*). |
| **C++** | Pentru citirea datelor de la consolă vom scrie :  cin>>var1>>var2>>…>>varn;  unde *var1*, *var2*, …, *varn* sunt variabile.  *Exemplu*  Fie date următoarele declaraţii :  **int** a;  **float** b;  Pentru citirea valorilor de la tastatură a acestor variabile într-o singură instrucţiune vom scrie :  cin>>a>>b; |

**Probleme pentru rezolvare independentă**

1. De la tastatură se citește un caracter. Elaboraţi un program prin intermediul la ecran se va afişa un triunghi de forma

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Date de intrare \* | Date de intrare \* | Date de intrare \* | Date de intrare \* |
| Date de ieşire  \*  \*\*  \*\*\*  \*\*\*\* | Date de ieşire  \* \* \* \*  \* \* \*  \* \*  \* | Date de ieşire  \*\*\*\*  \*\*\*  \*\*  \* | Date de ieşire  \*  \*\*  \* \* \*  \* \* \* \* |

1. Într-o tabără numărul de băieţi este cu 10 mai mare decât cel al fetelor. De la tastatură se citeşte numărul de fete. Elaboraţi un program prin intermediul căruia se va determina numărul elevilor din tabără. Exemplu: date de intrare: 50 date de ieşire: 110.
2. Într-un autobuz care pleacă în excursie sunt 7 copii. De la încă două şcoli urcă alţi copii, numărul acestora citindu-se de la tastatură. Elaboraţi un program prin intermediul căruia se va determina numărul copiilor care au plecat în excursie. Exemplu: Date de intrare: 15 20 Date de ieşire: 42 copii.
3. Un brăduţ este împodobit cu globuleţe albe, roşii şi albastre. Numărul globuleţelor albe se citeşte de la tastatură. Elaboraţi un program prin intermediul căruia se va determina numărul globuleţelor din brăduţul, dacă numărul de globuleţelor roşii este cu 3 mai mare decât numărul de globuleţe albe, iar numărul globuleţele albastre este mai mic cu 2 decât totalul celor albe şi roşii. Exemplu: Date de intrare: 12 Date de ieşire: 52.
4. Ion şi 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. Elaboraţi un program prin intermediul căruia la ecran se vor afişa numerele găsite de către Vasile. Exemplu: Ion spune 10, Vasile spune 8 9 10 11 12.
5. Doi copii au primit acelaşi n mere, n se citeşte de la tastatură. Elaboraţi un program prin intermediul căruia la ecran se va afişa numărul de mere pe care îl au copii dacă primul copil mănâncă un măr şi dă unul celuilalt copil. Exemplu: Date de intrare: 10 Date de ieşire: primul copil 8 mere al doilea copil 11 mere.
6. Maria vrea să verifice dacă greutatea şi înălţimea ei corespund vârstei pe care o are. Ea a găsit într-o carte următoarele formule de calcul ale greutăţii şi înălţimii unui copil, v fiind vârsta: greutate=2\*v+8 (în kg), înălţime=5\*v+80 (în cm). Elaboraţi un program prin intermediul căruia se va determina greutatea şi înălţimea ideală a unui copil, dacă vârsta se citeşte de la tastatură.
7. Se introduc de la tastatură trei cifre. Elaboraţi un program prin intermediul căruia se va afişa pe aceeaşi linie 5 numere formate cu aceste cifre luate o singură dată. Exemplu: date de intrare: 3 4 2 Date de ieşire: 324 342 243 234 432.
8. De la tastatură se citesc tei numere întregi. Elaboraţi un program prin intermediul căruia se vor determina toate sumele posibile de câte două numere. Afişarea va cuprinde şi termenii sumei, nu numai valoarea ei. Exemplu: Date de intrare: 2 13 4 Date de ieşire: 2+13=15 2+4=6 13+4=17.
9. De la tastatură se citeşte un număr întreg n(1≤n≤10). Elaboraţi un program prin intermediul căruia se va afişa la ecran tabla înmulţirii cu numărul n. Exemplu: pentru n=5, se va afişa pe verticală 1x5=5 2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45 10x5=50.
10. De la tastatură se citeşte un număr întreg, care reprezintă numărul de ani. Elaboraţi un program prin intermediul căruia la ecran se va afişa numărul de luni, zile şi ore corespunzătoare. Se consideră că un an are 365 zile. Exemplu: date de intrare: 2 date de ieşire: 24 luni 730 zile 17520 ore.
11. De la tastatură se citesc două numere întregi a şi b. Elaboraţi un program prin intermediul căruia să se schimbe între ele valorile variabilelor a şi b. La ecran se vor afişa noile valori ale lui a şi b.
12. \* Rezolvaţi problema **12)** şi fără a utiliza o variabilă intermediară.
13. Distanţa dintre două oraşe A şi B este de x km. Un şofer cu automobilul, parcurge această distanţă în a minute. Elaboraţi un program prin intermediul căruia la ecran se va afişa viteza medie a automobilului, dacă valorile lui x şi a se citesc de la tastatură.
14. Distanţa dintre două oraşe A şi B este de x km. Un şofer cu automobilul parcurge această distanţă. Elaboraţi un program prin intermediul căruia se va determina costul deplasării, dacă la distanţa de 100 km automobilul consumă în medie y litri de combustibil. Costul unui litru de combustibil, distanţa dintre oraşe şi consumul la 100km, se citesc de la tastatură.
15. O navă parcurge distanţa d (în km) dintre două staţii orbitale în a săptămâni şi b zile. Elaboraţi un program prin intermediul căruia se va determina viteza navei exprimată în km/oră.

|  |  |  |
| --- | --- | --- |
| **§2** |  | **Funcţii 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* *ex* | *exp(x);* |
| logaritmul natural *lnx* | *ln(x);* |
| Rădăcina pătrată a lui *x* | *sqrt(x);* |
| Valoarea absolută a lui *x* | *abs(x);* | Coincide cu tipul lui *x* |
| Pătratul lui *x* | *sqr(x);* |
| Rotunjirea lui *x* | *round(x)* | *real* | *integer* |
| Trunchierea lui *x* | *trunc(x)* |
| Paritatea lui *x,* True, daca 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** |
| *double asin( double arg ); double acos( double arg );* | Calculează arcsinusul/arccosinusul valorii **arg**; rezultatul este măsurat în radiani |
| *double atan( double arg ); double atan2( double y, double x );* | Calculează arctangenta valorii **arg**, respectiv a fracției **y/x** |
| *double floor( double num );* | Întoarce cel mai mare întreg mai mic sau egal cu **num** (partea întreagă inferioară) |
| *double ceil( double num );* | Întoarce cel mai mic întreg mai mare sau egal cu **num** (partea întreagă superioară) |
| *double sin( double arg ); double cos( double arg ); double tan( double arg );* | Calculează sinusul/cosinusul/tangenta parametrului **arg**, considerată în radiani |
| *double sinh( double arg ); double cosh( double arg ); double tanh( double arg );* | Calculează sinusul/cosinusul/tangenta hiperbolică a parametrului **arg** |
| *double exp( double arg );* | Întoarce valoarea *earg* |
| *double pow(double base, double exp );* | Întoarce valoarea *basearg* |
| *double log( double num );* | Calculează logaritmul natural (de bază **e**) al valorii **arg** |
| *double log10( double num );* | Calculează logaritmul în baza 10 al parametrului |
| *double sqrt( double num );* | Calculează rădăcina pătrată a parametrului |
| *double fmod( double x, double y );* | Întoarce restul împărțirii lui **x** la **y** |
| *double fabs( double arg );* | Întoarce valoarea absolută a lui **arg** |

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

**Probleme pentru rezolvare independentă**

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

1. ;
2. ;
3. ;
4. ;
5. ;
6. ;
7. ;
8. ;
9. ;
10. .

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

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

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 

|  |  |
| --- | --- |
| **§3** | **Instrucţiunea de ramificare if** |

|  |  |
| --- | --- |
| ***Instrucţiunea de ramificare if*** | |
| **Pascal** | 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’)** |
| **C/C++** | **if**...**else**  este cea mai simplă instrucţiune condiţională.  Poate fi folosită în mai multe forme:  **if**( condiţie ){ // instrucţiuni }  **if**( condiţie ){ //instrucţiuni }  **else**{ //alte instrucţiuni }  **if**( condiţie1 ){//instrucţiuni1 }  **else** **if**( condiţie2 ){ //instrucţiuni2 }  **else** **if**( condiţie N ){ //instrucţiuni N } |

**Probleme pentru rezolvare independentă**

1. Se introduc punctajele a doi sportivi. Elaboraţi un program prin intermediul căruia se va afişa punctajele în ordine descrescătoare. Exemplu: Date de intrare 100 134 Date de ieşire: 134 puncte 100 puncte
2. Se dau două numere. Elaboraţi un program prin intermediul căruia se va înmulţi cel mai mare cu doi şi cel mai mic cu trei şi să se afişeze rezultatele. Exemplu: date de intrare: 3 7 date de ieşire: 9 14
3. Se introduc două numere întregi. Elaboraţi 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 ieşire: Nu; date de intrare: 5 6 date de ieşire: Da.
4. Ionel spune părinţilor doar notele mai mari sau egale cu 7. Într-o zi el a luat trei note. Elaboraţi un program prin intermediul căruia la ecran se vor afişa doar notele pe care le va comunica Ionel părinţilor. Exemplu : Date de intrare 8 7 5 Date de ieşire 8 7.
5. Se introduc două numere. Elaboraţi un program prin intermediul căruia se va afişa câtul dintre primul şi al doilea, dacă al doilea număr este diferit de 0 sau mesajul “Împărţire imposibilă” în caz contrar. Exemple: Date de intrare 10 3 Date de ieşire 3.33 Date de intrare 45 0 date de ieşire Impartire imposibila.
6. Un ascensor pentru copii acceptă o greutate de maxim 100 kg. De la tastatură se introduc greutăţile a doi copii. Elaboraţi un program prin intermediul căruia se va afişa mesajul POT INTRA AMBII COPII, dacă greutatea copiilor nu depăşeşte 100 kg şi mesajul INTRA PE RIND în caz contrar. Exemple: Date de intrare: greutăți copii 87 50 Date de ieşire : Intra pe rând Date de intrare 45 52 Date de ieşire 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. Elaboraţi 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 ieşire: Va fi pedepsit.
8. Elaboraţi un program prin intermediul căruia se va determină greutatea ideală a unei persoane cunoscând înălţimea, vârsta şi sexul persoanei. Formulele de calcul sunt:

*Gmasculin = 50 + 0.75 \* (inaltime - 150) + (varsta - 20) / 4,*

*Gfeminin = Gmasculin – 10,*

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

1. De la tastatură se citesc trei numere întregi. Elaboraţi 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 ieşire: 45 impar 3 impar 24 par.
2. Într-o tabără, băieţii sunt cazaţi câte 4 într-o căsuţă, în ordinea sosirii. Ionel a sosit al n-lea, n se citește de la tastatură. Elaboraţi un program prin intermediul căruia se va determina în a câta căsuţă se va afla Ionel Exemplu : date de intrare : n=69 date de ieşire : casuta 17.
3. Elevii clasei a V-a se repartizează în clase câte 25 în ordinea mediilor clasei a IV-a. Radu este pe locul x (1<=X<=125) în ordinea mediilor, x se citește de la tastatură. Elaboraţi 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 ieşire : C.
4. De la tastatură se citesc trei numere diferite. Elaboraţi un program prin intermediul căruia:
   * 1. se va afişa cel mai mare şi cel mai mic număr. Exemplu : Date de intrare 45 34 78 Date de ieşire max=78 min=34. Numerele vor fi afișate în ordine crescătoare. Exemplu : Date de intrare 4 2 6 Date de ieșire 2 4 6.
     2. se va afişa numărul a cărei valoare este cuprinsă între valorile celorlalte două. Exemplu: Date de intrare 12 14 10 Date de ieşire 12.
     3. se va verifica dacă numerele introduse, formează o secvenţă de numere consecutive. Exemple: Date de intrare 3 4 5 Date de ieşire Da Date de intrare 4 5 7 Date de ieşire Nu.
5. Andrei primeşte într-o zi trei note, nu toate bune. Se hotărăşte ca, dacă ultima notă este cel puţin 8, să le spună părinţilor 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. Elaboraţi un program prin intermediul căruia se vor afişa notele pe care Andrei le va comunica părinţilor. Exemple : Date de intrare 6 9 9 Date de ieşire 6 9 9 ; Date de intrare 8 5 7 Date de ieşire 8.
6. De la tastatură se citesc trei numere. Elaboraţi un program prin intermediul căruia se va afișa numărul mai mare dintre al doilea şi 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 ieşire 100 ; Date de intrare 34 -25 10 Date de ieşire 9.
7. Să se afişeze cel mai mare număr par dintre doua numere introduse în calculator. Exemple : Date de intrare 23 45  Date de ieşire nu exista numar par ; Date de intrare 28 14 Date de ieşire 28 ; Date de intrare 77 4 Date de ieşire 4.
8. De la tastatură se citesc trei cifre diferite de 0. Elaboraţi un program prin intermediul căruia se va afişa numărul format din aceste cifre, astfel încât acesta să fie maximal. Exemplu: Date de intrare : 4 5 3 Date de ieşire: 543.
9. De la tastatură se citesc trei cifre. Elaboraţi un program prin intermediul căruia se va afişa numărul format din aceste cifre, astfel încât acesta să fie minimal. Exemplu: Date de intrare : 4 5 3 Date de ieşire: 345. Notă: Participă şi cifra 0, dar numărul nu poate începe cu 0.

|  |  |
| --- | --- |
| **§4** | **Instrucţiunea cu selecţie multiplă** |

|  |  |
| --- | --- |
| ***Instrucţiunea cu selecție multiplă*** | |
| **Pascal** | Case a of  cazul 1 :instrucțiunea 1;  cazul 2 :instrucțiunea 2;  …  cazul n :instrucțiunea n  [ else instrucțiunea n+1; ]  end;  Menționăm, că:   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 *a* se regăsește în unul din cazuri – se execută instrucțiunea respectivă și se trece la următoarea instrucțiune, după end. |
| **C/C++** | **switch** este o instrucţiune menită să simplifice structurile condiţionale cu mai multe condiţii.  **switch**( expresie ){  **case** constanta1: //instrucţiuni1 **break**;  **case** constanta2: //instrucţiuni2 **break**;  **[default**: //instrucţiuni]  }  Valoarea expresie de tip ordinal este evaluată la un tip întreg, apoi această valoare este comparată cu fiecare constantă;  este rulat blocul de instrucţiuni 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. |

**Probleme pentru rezolvare independentă**

1. De la tastatură se citeşte un caracter (literă). Elaboraţi un program care va afişa textul E VOCALA, dacă caracterul introdus este vocală. De exemplu: date de intrare E, date de ieşire E VOCALA şi NU E VOCALA în caz contrar.
2. De la tastatură se citeşte o cifră. Elaboraţi un program care va afişa cifra scrisă cu caractere. De exemplu: date de intrare 5, date de ieşire cinci
3. De la tastatură se citesc 2 numere întregi şi un operator (+ - \* / %). Elaboraţi un program care în dependenţă de datele introduse va afişa 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 şi un operator (+ - \* /). Elaboraţi un program care în dependenţă de datele introduse va afişa 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 citeşte un număr pozitiv, mai mic sau egal cu 100. Elaboraţi un program care va afişa numărul citit cu cifre romane. De exemplu: date de intrare 18, date de ieşire XVIII
6. Magazinul comercial Nr 1, oferă clienţilor 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, Marţi 5%, Miercuri 6%, Joi 7%, Vineri, 8% Sâmbătă 9%, Duminică 10%. Elaboraţi un program care în dependenţă de ziua şi suma introdusă va afişa la ecran reducerea oferită, cât şi Suma ce necesită a fi achitată. De exemplu: date de intrare Duminică 1000, date de ieşire reducere 100 lei, suma ce necesită a fi achitată 900 lei.

|  |  |
| --- | --- |
| **§5** | **Instrucţiuni ciclice** |

|  |  |
| --- | --- |
| În varianta **Pascal** | În varianta **C/C++** |
| **while** execută un bloc de instrucţiuni atât timp cât o anumită condiţie este adevărată. Forma generală a unui ciclu while este: | |
| **while** (expresie) do instrucţiune;  **while** (expresie) do  begin instrucţiuni end; | **while** (expresie){  //instrucţiuni } |
| Dacă condiţia din expresie este adevărată (are o valoare nenulă), instrucţiunile din blocul de după while sunt executate. expresie este reevaluată după fiecare iteraţie. Pentru ca condiția să devină falsă este în responsabilitatea programatorului. | |
| *Repeat …..until* (expresie); | **do** ... **while** (expresie); |
| este o instrucţiune repetitivă similară cu cea precedentă, singura diferenţă fiind că expresie este evaluată după executarea instrucţiunilor, nu înainte. Astfel, blocul va fi executat cel puţin o dată. | |
| *Repeat*  *Instrucțiuni*  *.until* (expresie); | **do**{  //instrucţiuni } **while** (expresie); |
| **for** reprezintă o formă mai simplă de a scrie un **while** însoţit de o expresie iniţială şi una finală. Deosebim 2 variante: una cu pasul +1:  *for i:=1 to 10 do* instrucțiune;  alta cu pasul -1  *for i:=10 downto 1 do* instrucțiune; | **for** reprezintă o formă mai simplă de a scrie un **while** însoţit de o expresie iniţială şi de o expresie de incrementare. Forma sa este:  **for**(expresie1;expresie2;expresie3){  //instrucţiuni } |
| Exemplul următor prezintă un ciclu cu funcţionalitate identică (tipărirea primelor 10 numere naturale), folosind cele 3 instrucţiuni repetitive: | |
| 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;  end. | **int** main(){  **int** i;  [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("Ciclu for\n");   **for**(i=1;i<=10;i++)  [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("i=%d\n", i);  [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("Ciclu while\n");  i=1;  **while** (i <= 10) {  [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("i=%d\n", i);  i++;  }  [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("Ciclu do while\n");  i=0;  **do**{  i++;  [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("i=%d\n", i);  }**while**(i < 10);  } |

**Probleme pentru rezolvare independentă**

1. Elaboraţi un program prin intermediul căruia se vor afişa toate numerele de forma a23a care se împart exact la 6.
2. De la tastatură se citesc două numere pozitive a şi n. Elaboraţi un program prin intermediul căruia se va afișa numărul a urmat de n zerouri. Exemplu: Date de intrare a=34 n=5 Date de ieşire 3400000.
3. De la tastatură se citesc numărul n, n(1≤n≤10).. Elaboraţi un program prin intermediul căruia se va afișa Să se afişeze tabla înmulţirii cu n. Exemplu: Date de intrare n=5 date de ieşire 1x5=5 2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45 10x5=50.
4. De la tastatură se citește un număr întreg pozitiv. Elaboraţi un program prin intermediul căruia se vor afișa toţi divizorii unui număr natural citit. Exemplu: Date de intrare 12 Date de ieşire 1 2 3 4 6 12.
5. De la tastatură se citește un număr întreg pozitiv. Elaboraţi un program prin intermediul căruia se va verifica dacă este numărul este prim. Exemple: Date de intrare 23 date de ieşire Este Prim Date de intrare 45 Date de ieşire Nu este prim.
6. Un ascensor parcurge distanţa dintre două etaje a şi b. Elaboraţi un program prin intermediul căruia se vor afişa toate etajele parcurse, în ordinea atingerii lor. Exemple: Date de intrare a=4 b=7 Date de ieşire 4 5 6 7; Date de intrare a=10 b= 8 Date de ieşire 10 9 8.
7. De la tastatură se citesc n numere întregi pozitive, fiind date numerele a, b şi c. Elaborați un program care să afişeze toate numerele care se divid cu a sau b şi sunt mai mici decât c.
8. De la tastatură se citeşte numărul n, întreg pozitiv. Elaboraţi un program prin intermediul căruia se vor determina sumele:

s1=1+2+…+n;

s2=1+3+…+2n-1;

s3=2+4+…+2n;

s4=1\*2+2\*3+3\*4+…+(n-1)\*n;

s5=1+1\*2+1\*2\*3+…+1\*2\*3\*…\*n;

s6=2-3+4-5+…-99+100;

1. Se citesc pe rând temperaturile medii ale fiecărei luni a unui an, ca numere întregi. Elaboraţi un program prin intermediul căruia se va afişa cu două zecimale media anuală a temperaturilor pozitive şi a celor negative. Exemplu: Date de intrare -5 -3 1 8 12 17 20 21 18 10 6 -2 Date de ieşire medie\_poz=13.66 medie\_neg=-3.33.
2. Se citesc mediile a n elevi, ca numere reale. Elaboraţi un program prin intermediul căruia se va afişa cea mai mare şi cea mai mică medie. Să se verifice dacă sunt corigenţi. Exemplu : Date de intrare n=4 9.50 4.25 9.66 6.33 Date de ieşire max=9.66 min=4.25 1 corigent.
3. Se dau două numere întregi nenule. Elaboraţi un program prin intermediul căruia se va afişa cmmdc şi cmmmc al lor. Exemplu : Date de intrare 12 32 Date de ieşire cmmdc=4 cmmmc 96.
4. Se dau trei numere. Elaboraţi un program prin intermediul căruia se va afişa cmmmdc al lor. Exemplu : Date de intrare 12 32 36 Date de ieşire 4.
5. Se citesc de la tastatură patru numere întregi diferite de zero. Numerele reprezintă în ordinea citirii: numărătorul şi numitorul primei fracţii, respectiv numărătorul şi numitorul celei de a doua fracţii. Elaboraţi un program prin intermediul căruia se va afişa suma celor două fracţii. Exemplu: Date de intrare: 6 18 12 24 Date de ieşire: 5 6.
6. De la tastatură se citesc 2 numere nenegative mai mici decât 10, care reprezintă lungimile laturilor unui dreptunghi. Elaboraţi un program care în dependenţă de datele introduse va afişa un dreptunghi. Exemplu:

|  |  |
| --- | --- |
|  |  |
| Date de intrare 4 3 | Date de intrare 4 3 |
| Date de ieşire  1111  2222  3333 | Date de ieşire  1234  1234  1234 |

1. Se introduc succesiv numere nenule întregi până la introducerea numărului 0. Să se afişeze suma tuturor numerelor pare introduse. Exemplu: Date de intrare 3 5 4 2 0 Date de ieşire 6.
2. Se introduc succesiv numere nenule întregi până la introducerea numărului 0. Să se afişeze suma tuturor numerelor divizibile cu 3 introduse. Exemplu: Date de intrare 3 5 6 2 0 Date de ieşire 9.
3. De la tastatură se citeşte un număr întreg nenegativ n, mai mic decât 10. Elaboraţi un program care va afişa un triunghi.

|  |  |
| --- | --- |
|  |  |
| Date de intrare 4 | Date de intrare 4 |
| Date de ieşire  1  12  123  1234 | Date de ieşire  1234  123  12  1 |

1. Se citesc numere de la tastatură până la introducerea unui număr impar divizibil cu 3. Să se afişeze suma tuturor numerelor pare şi a numerelor impare introduse. Precizaţi câte numere au fost introduce. Exemplu: Date de intrare 7 4 5 2 1 9 Date de ieşire suma pare =6 suma impara=23.
2. Se citesc numere de la tastatură până la introducerea unui număr divizibil cu 5. Să se afişeze suma tuturor numerelor pare şi a numerelor impare introduse. Precizaţi câte numere au fost introduce. Exemplu: Date de intrare 7 4 6 2 1 25 Date de ieşire suma pare =12 suma impara=35.
3. 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. Elaboraţi un program prin intermediul căruia se va afişa după câte zile Făt-frumos o eliberează pe Ileana Cosânzeana. Exemplu: Date de intrare x=10 a=4 b=1 Date de ieşire 3 zile.
4. Se citesc de la tastatură numere naturale până când suma numerelor pare este mai mare decât k. Câte numere au fost introduse şi care este suma numerelor pare? Exemplu: Date de intrare: K=12 5 8 1 2 2 3 6 Date de ieşire: 7 (numere) 18 (suma celor pare).
5. Se citesc de la tastatură numere naturale până când suma numerelor pare este mai mare decât k. Câte numere au fost introduse şi care este suma numerelor impare? Exemplu: Date de intrare: K=12 5 8 1 2 2 3 6 Date de ieşire: 7 (numere) 9 (suma celor impare)
6. Se citesc de la tastatură numere întregi pozitive atât timp cât suma lor nu depăşeşte 1000. Să se scrie un program care să afişeze cea mai mică şi cea mai mare valoare a acestor numere. Exemplu: 550 345 100 45 Date de ieşire max=550 min=100.
7. 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 afişeze cea mai mică şi cea mai mare valoare a acestor numere. Exemplu: 7 3 6 2 Date de ieşire max=7 min=2.
8. De la tastatură se citesc n numere întregi pozitive. Elaboraţi un program care va determina suma şi media numerelor citite cu excepţia numerelor unde suma cifrelor este un număr impar.
9. De la tastatură se citesc n numere întregi pozitive. Elaboraţi un program care va determina suma şi media numerelor citite cu excepţia numerelor unde suma cifrelor cărora este un număr divizibil cu trei.
10. De la tastatură se citesc datele despre n (n>2) dreptunghiuri (lungimea şi lăţimea). Elaboraţi un program care va afişa dreptunghiul (lungimile laturilor, aria şi perimetru) cu suprafaţa maximă şi dreptunghiul cu cel mai mic perimetru.
11. De la tastatură se citesc datele despre n (n>2) triunghiuri dreptunghice (lungimea catetelor). Elaboraţi un program care va afişa triunghiul (lungimile laturilor aria şi perimetru) cu suprafaţa maximă şi triunghiul cu cea mai mică ipotenuză.
12. Elaboraţi un program care va afişa toate numerele întregi pozitive a, b, c, mai mici ca 20, cu următoarea proprietate: a2+b2=c2
13. Elaboraţi un program care va afişa toate numerele întregi pozitive a, b, c, mai mici ca 20, cu următoarea proprietate: (a+b)3=c3
14. De la tastatură se citesc trei numere întregi pozitive a, b, c mai mici ca 100000. Să se afişeze toţi divizorii comuni.
15. De la tastatură se citesc trei numere întregi pozitive a, b, c mai mici ca 10000. Să se afişeze la ecran:
    1. numărul care are suma cifrelor o valoare maximă;
    2. numărul care are produsul cifrelor o valoare maximă;
    3. multiplu comun ale acestor numere.

|  |  |
| --- | --- |
| **§6** | **Recapitulare** |

1. Se consideră numărul întreg ***N.*** Scrieţi un program care determină numărul de cifre impare şi numărul de cifre pare ale acestui număr (zero considerându-se număr par).
2. Se consideră ***N*** un număr întreg. Scrieţi 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 ieşire va fi afişat cuvântul* ***Da***

şi dacă*N la Intrare este* ***275*** *la ieşire va fi afişat cuvântul* ***Nu***

1. Scrieţi un program care să determine toate numerele naturale mai mici decât **MAXINT** cu proprietatea că sunt egale cu suma factorialelor cifrelor lor.

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

1. Scrieţi un program care determină toate numerele naturale mai mari ca 1 şi mai mici decât un număr natural diferit de zero dat şi 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 afişează la ecran, pe primul loc – numărul ce indică cantitatea de numere cu proprietatea dată, şi în continuare – numerele găsite.

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

1. Un număr de trei cifre se numeşte *„norocos”* dacă suma cărorva două cifre ale acestui număr este egală cu a treia. Elaboraţi un program care citeşte de la tastatură n numere, formate din trei cifre şi determină numărul de numere *„norocoase”*
2. Un număr de patru cifre se numeşte *„norocos”* dacă suma primelor două cifre este egală cu suma ultimelor două. Elaboraţi un program care citeşte de la tastatură n numere, formate din patru cifre şi determină numărul de numere *„norocoase”*
3. Un număr de patru cifre se numeşte *„nenorocos”* dacă suma primelor două cifre este egală cu suma ultimelor două şi este *13*. Elaboraţi un program care citeşte de la tastatură n numere, formate din patru cifre şi determină numărul de numere *„nenorocoase”*
4. Scrieţi un program care citeşte de la tastatură numărul natural ***N*** şi determină numărul minim care poate fi obţinut 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. Elaboraţi un program prin intermediul căruia se vor efectua următoarele operaţii:
2. Se vor afişa la ecran numerele la un interval de 5 poziţii;
3. Se vor afişa la ecran numerele în ordinea inversă a introducerii lor la un interval de 4 poziţii;
4. Se vor afişa doar numerele pare;
5. Se vor afişa doar numerele impare;
6. Se vor afişa doar numerele mai mari ca x şi nu sunt divizibile cu y, x şi y se citesc de la tastatură;
7. Se vor afişa doar numerele mai mari ca x şi mai mici decât y, x şi y se citesc de la tastatură;
8. Se vor afişa poziţiile numerelor negative şi impare din tablou;
9. Se va afişa media numerelor pare;
10. Se vor afişa poziţiile numerelor formate din 2 cifre;
11. Se va afişa suma numerelor divizibile cu 3;
12. Se va afişa cele mai mari 2 numere;
13. Se va afişa toate numerele cu excepţia celui minimal şi maximal.
14. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program care va afişa elementele introduse, apoi va mari fiecare element al tabloului cu 10. Elementele modificate se vor afişa la ecran.
15. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program prin intermediul căruia elementele pare ale vectorului, se vor dubla, iar cele impare se vor tripla. Se vor afişa elementele vectorului modificat.
16. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program prin intermediul căruia primul element din tablou va fi înlocuit cu elementul minimal şi invers. Se vor afişa elementele vectorului modificat.
17. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program prin intermediul căruia al treilea element va fi înlocuit cu elementul maximal şi invers. Se vor afişa elementele vectorului modificat.
18. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program prin intermediul căruia se va crea un nou vector care va fi format numai din elementele vectorului iniţial care au exact cel puţin trei divizori. Se vor afişa elementele ambilor vectori.
19. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program care va afişa elementele introduse, apoi va micşora fiecare element al tabloului cu 12. Elementele modificate se vor afişa la ecran.
20. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program prin intermediul căruia:
21. elementele negative ale vectorului, se vor dubla, iar cele mai mari decât 10 se vor micşora cu 15. Se vor afişa elementele vectorului modificat.
22. se va crea un nou vector care va fi format numai din elementele impare ale vectorului iniţial. Se vor afişa elementele ambelor vectori.
23. care va afişa elementele introduse, apoi se va mari fiecare element al tabloului de 5 ori. Elementele modificate se vor afişa la ecran.
24. elementele impare ale vectorului, se vor mări cu 25, iar cele negative micşora de 2 ori. Se vor afişa elementele vectorului modificat.
25. elementul maximal al vectorului, va fi înlocuit cu cel minimal şi reciproc. Se vor afişa elementele vectorului modificat.
26. De la tastatură se citesc elementele unui vector de dimensiune n (n<1000) cu numere întregi. Elaboraţi un program prin intermediul căruia se va crea un nou vector care
27. va fi format numai din elementele pare ale vectorului iniţial. Se vor afişa elementele ambilor vectori.
28. va fi format numai din elementele vectorului iniţial care au doi sau patru divizori. Se vor afişa elementele ambilor vectori.
29. va fi format numai din elementele divizibile cu 3 ale vectorului iniţial. Se vor afişa elementele ambilor vectori.
30. va fi format numai din elementele vectorului iniţial care au cel mult patru divizori. Se vor afişa elementele ambilor vectori.

|  |  |
| --- | --- |
| **§2** | **Tablouri bidimensionale** |

1. De la tastatură se citesc elementele unui tablou bidimensional cu n – linii şi m – coloane n,m<10 de numere întregi. Elaboraţi un programa care:
2. Va afişa elementele matricei;
3. Va determina numărul elementelor pare, mai mici decât 10;
4. Va determina numărul elementelor formate dintr-o singură cifră;
5. Va înlocui elementele divizibile cu 3, cu valoarea numărului maximal \*3, şi va afişa matricea modificată;
6. Va înlocui elementele pare cu valoarea numărului maximal \* 2, şi va afişa matricea modificată;
7. Va afişa elementele de pe coloana 2 şi elementele de pe linia 1;
8. Va afişa elementele de pe coloana 1 şi elementele de pe linia 2;
9. Va afişa elementele de pe coloana a şi elementele de pe linia b.
10. Să se construiască un tablou pătratic de dimensiune n,n cu primele n\*n numere:
11. pare. Exemplu: n=3 se va afişa

2 4 6

8 10 12

14 16 18

1. impare. Exemplu: n=3 se va afişa

1 3 5

7 9 11

13 15 19

1. divizibile cu 5. Exemplu: n=3 se va afişa

5 10 15

20 25 30

35 40 45

1. De la tastatură se citesc elementele unui tablou bidimensional cu n linii şi m coloane. Elaboraţi un program care va afişa la ecran elementele matricei, va adăuga o linie nouă, elementele cărora vor fi:
   * + - 1. maximul de pe fiecare coloană. Rezultatul se va afişa la ecran. Exemplu Pentru o matrice cu 3 lini şi 3 coloane

1 9 3

8 5 6

7 8 19

Linia nouă adăugată va fi 8 9 19

* + - * 1. suma elementelor de pe coloana respectivă. Rezultatul se va afişa la ecran. Exemplu Pentru o matrice cu 3 lini şi 3 coloane

1 2 3

4 5 6

7 8 9

Linia nouă adăugată va fi 12 15 18

* + - * 1. suma dintre elementele primei linii şi celei dea doua linie. Rezultatul se va afişa la ecran. Exemplu Pentru o matrice cu 2 lini şi 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.

1. De la tastatură se citesc elementele unui tablou bidimensional cu n linii şi n coloane. Elaboraţi un program care va afişa la ecran elementele tabloului şi:
2. va mări elementele de pe diagonala principală cu 5. Rezultatul se va afişa la ecran;
3. va afişa elementele de pe linia ce conţine elementul minimal;
4. va mări elementele unei coloane cu 15. Numărul de ordine a coloanei se citeşte de la tastatură. Rezultatul se va afişa la ecran.
5. De la tastatură se citesc elementele unui tablou bidimensional cu n linii şi m coloane. Elaboraţi un program care va afişa la ecran elementele tabloului şi va mări elementele unei linii cu 10. Numărul de ordine a liniei se citeşte de la tastatură. Rezultatul se va afişa la ecran.
6. De la tastatură se citesc elementele unui tablou bidimensional cu n linii şi m coloane. Elaboraţi un program care va afişa la ecran elementele tabloului şi va afişa elementele de pe linia suma elementelor cărora este maximă.
7. De la tastatură se citesc elementele unui tablou bidimensional cu n linii şi m coloane. Elaboraţi un program care va afişa la ecran elementele tabloului şi va afişa elementele de pe coloana ce conţine elementul maximal.

|  |  |
| --- | --- |
| **§3** | **Şiruri de caractere** |

**Funcţii și proceduri** *de prelucrare a şirurilor de caractere*

În varianta **Pascal**

|  |  |
| --- | --- |
| **Antet** | **Descriere** |
| writeln(s); | extrage la ecran şirului s |
| readln(s); | citește şirul de la tastatură în variabila s |
| LENGTH(S) | Lungimea şirului S |
| CONCAT(s1[,s2,…,sn]) | Concatenează şirurile S1, S2, …Sn. Echivalentă cu S1+S2+…+Sn. |
| POS(Subşir,Şir) | Obţinem poziţia apariţiei subşirului în şir. Dacă rezultatul e 0 rezultă că aşa subşir nu există în şir. |
| COPY (Şirul, De\_La\_Care\_Poziție, Câte\_Carectere) | Obţinem un subşir, din şirul dat, începând de la care poziţie, de lungimea Câte\_Carectere. |
| INSERT(Ce\_Inseram, In\_Care\_Şir, De\_la\_care\_Poziţie) | Se inserează un subşir într-un şir, începând cu poziţia indicată. |
| DELETE (Şirul, De\_La\_Care\_Poziție, Câte\_Carectere) | Din şirul dat se exclud de la poziţia indicată numărul indicat de caractere. |
| VAL(Şir, Variabilă\_Numerică, Cod) | Transformă şirul de caractere în număr. Dacă nu este posibil – cod este diferit de zero, şi dacă s-a transformat cu succes – Cod este 0. |
| STR(Număr[:m[:n]]; Şir) | Transformă numărul în şir 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 conţinutul şirului sir |
| gets(sir); | citește caractere de la tastatură în variabila sir |
| ***int*** *strcmp*(***char*** *\*s1,* ***char*** *\*s2*); | compară şirurile *s1* şi *s2* returnează o valoare pozitivă, dacă şirul *s1>s2*, 0 dacă *s1* este egal cu *s2* şi o valoare negativă dacă *s1<s2* |
| ***int*** *strcmpi*(***char*** *\*s1,* ***char*** *\*s2*); | similară cu *strcmp* majuscule sunt nesemnificative |
| ***int*** *strncmp*(***char*** *\*s1,* ***char*** *\*s2, int n*); | similară cu *strcmp*, dar se compară şirurile *s1* şi *s2* pentru cel mult *n* caractere din fiecare şir |
| ***char*** *\*strcpy*(***char*** *\*d,* ***char*** *\*s*); | copiază şirul sursă *s* în şirul destinaţie *d* şi returnează adresa şirului destinaţie |
| ***char*** *\*strncpy*(***char*** *\*d,* ***char*** *\*s,* ***int*** *n*); | copiază maxim *n* caractere de la sursă *s* la destinaţia *d* şi returnează adresa şirului destinaţie |
| ***int*** *strlen*(***char*** *\*s*); | returnează lungimea şirului fără a număra caracterul terminator ’\0’ |
| ***char*** *\*strcat*(***char*** *\*d,* ***char*** *\*s*); | concatenează cele două şiruri şi returnează adresa şirului rezultat |
| **double** atof(sir); | converteşte un şir către tipul **double**. Dacă această conversie eşuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0 |
| **long** **double** \_atold(sir); | converteşte un şir către tipul **long double**. Dacă această conversie eşuează, valoarea întoarsă este 0 |
| **int** atoi(sir); | converteşte un şir către tipul **int**. Dacă această conversie eşuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0 |
| **long** atol(sir); | converteşte un şir către tipul **long**. Dacă aceasta conversie eşuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0. |
| itoa(**int** valoare,sir,**int** baza); | converteşte o valoare de tip **int** în şir, care este memorat în variabila sir. baza reţine baza de numeraţie către care să se facă conversia |
| ltoa(**long** valoare,sir,**int** baza); | converteşte o valoare de tip **long int** în şir, care este memorat în variabila sir |
| ultoa(unsigned **long** valoare,sir,**int** baza); | converteşte o valoare de tip unsigned long în şir, care este memorat in variabila şir |

**Probleme pentru rezolvare independentă**

1. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va determina:
2. numărul de majuscule din şir;
3. numărul de cifre din şir;
4. numărul de caractere speciale(paranteze, operatori aritmetici) din şir;
5. numărul de cifre pare citite.
6. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va înlocui:
   * + 1. cifrele din şir cu semnul +;
       2. vocalele din şir cu spaţiu;
       3. majuscule din şir cu cifra 8.
7. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va afişa:
   * 1. doar cifrele din şir;
     2. fiecare cuvânt din rând nou;
     3. doar consoanele din şir;
     4. toate cifrele cu excepţia celor divizibile cu 3.
8. De la tastatură se citesc mai multe cifre. Elaboraţi un program prin intermediul căruia se va determina dacă măcar un număr care poate fi format din cifrele citite este par.
9. De la tastatură se citesc patru cuvinte, fiecare cuvânt fiind citit într-o singură variabilă. Elaboraţi un program care va forma o frază, va include toate cuvintele în şir. Fiecare cuvânt va fi despărţit prin spaţiu, ultimul caracter din frază va fi semnul punct. La ecran se vor afişa cuvintele citite cât şi fraza formată.
10. De la tastatură se citesc patru cuvinte, fiecare cuvânt fiind citit într-o singură variabilă, un cuvânt va fi format din minim 3 caractere. Elaboraţi 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 şi n/2 caractere din cuvântul patru(n – lungimea cuvântului). La ecran se vor afişa cuvintele citite cât şi cuvântul format.
11. De la tastatură se citesc patru cuvinte, fiecare cuvânt fiind citit într-o singură variabilă. Elaboraţi un program care va afişa cuvintele în ordine alfabetică. Fiecare cuvânt va fi afişat din rând nou.

|  |  |
| --- | --- |
| **§4** | **Fişiere** |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 *var* | 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 dispare. | 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) | |
| Citirea datelor din fișierul textual poate fi organizată în mai multe moduri, în dependență de condițiile problemei.  *Exemplu 1*: Fișierul *tablou.txt* conține în prima linie 2 numere întregi – *m* și *n*, care reprezintă numărul de linii și coloane al unui tablou (ambele mai mici ca 100). În următoarele *m* rânduri sunt scrise câte *n* numere reale – elementele fiecărei linii.  *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);*  *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 şi 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 fişierului se va utiliza funcţia fopen. Mod de utilizare :  FILE \*fopen(adresa\_file, mod\_de\_deschidere);  Deschide fişierul cu numele dat pentru acces de tip mod\_de\_deschidere. Returnează pointer la fişier sau NULL dacă fişierul nu poate fi deschis, valoarea returnată este memorată în variabila fişier, care a fost declarată pentru accesarea lui.  adresa\_file indică numele şi adresa fişierului. Pentru a deschide fişierul *azi.txt* în calitate de adresă vom indica “*azi.txt*”, dacă fişierul este pe discul c în mapa temp atunci vom scrie “*C: \\temp\\azi.txt*”.  Parametrul mod\_de\_deschidere este un şir de caractere care poate avea următoarele valori :  *"r"* - readonly , este permisă doar citirea dintr-un fişier existent;  *"w"* - write, creează un nou fişier, sau dacă există deja, distruge vechiul conţinut;  *"a"* - append, deschide pentru scriere un fişier existent (scrierea se va face în continuarea informaţiei deja existente în fişier, deci pointerul de acces se plasează la sfârșitul fişierului );  *"+"* - permite scrierea şi citirea - actualizare (ex: "r+", "w+", "a+"). O citire nu poate fi direct urmata de o scriere şi reciproc. Întâi trebuie repoziţionat cursorul de acces printr-un apel la *fseek*.  *"b"* - specifică fişier de tip binar.  **Funcţii de prelucrare a fişierelor**   * **int** feof(FILE \*fp); returnează o valoare nenulă dacă s-a întâlnit sfârșitul de fişier la ultima operaţie de intrare şi 0 în caz contrar; * **int** fclose(FILE \*fp); unde fp este pointerul returnat de funcţia *fopen*. Prin închiderea fişierului se eliberează zona tampon alocată acestuia. Funcţia returnează valoarea 0 dacă operaţia de închidere s-a efectuat cu succes; * **int** fsetpos(FILE \*fp, **long** **int** \*poz); atribuie indicatorului valoarea variabilei indicată prin pointerul *poz* şi returnează valoarea 0 în caz de succes; * **int** fseek(FILE \*fp, **long** offset, **int** whence); repoziţionează indicatorul fişierului la valoarea *whence+offset*; *whence* poate avea următoarele valori:   SEEK\_SET= 0 - început de fişier;  SEEK\_CUR= 1 - poziţie curentă;  SEEK\_END= 2 - sfârșit de fişier.   * **int** getc(FILE \*fp); returnează următorul caracter citit din fişierul fp sau EOF dacă se întâlnește sfârșitul fişierului; * **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); citeşte maxim *n-1* caractere din fişierul fp sau până la '\n' inclusiv, le depune în *s*, adaugă la sfârșit '\0' şi returnează adresa şirului *s*, în caz de eroare întoarce valoarea NULL; * **int** fputs(char \*s, FILE \*fp); scrie şirul *s* în fişier, fără caracterul '\0', în caz de eroare întoarce valoarea EOF; * **int** fscanf(FILE \*fp, **char** \*format [, adresa, ...]); citeşte date din fişierul fp, conform formatului specificat (specific funcţiei scanf); * **int** fprintf(FILE \*fp, **char** \*format [, argum...]); scrie date, conform formatului (specific funcţiei printf) în fişierul fp, returnează numărul de octeţi transferaţi sau EOF în caz de eroare. | | |
| **C++** | Un fişier este văzut în C++ ca un obiect, deci ca o variabilă de un tip clasă. Se pot folosi 3 clase predefinite în biblioteca *fstream.*h :   * *fstream* pentru fişiere ce pot fi folosite în citire sau în scriere; * *ifstream* pentru fişiere din care este permisă doar citirea; * *ofstream* pentru fişiere in care este permisă doar scrierea.   Declararea unei variabile de tip fişier :  nume\_clasa nume\_file;  Deschiderea fişierului se efectuează prin intermediul metodei *open*. Mod de utilizare :  nume\_file.open(adresa\_file, mod\_de\_deschidere);  Argumentul mod\_de\_deschidere este opţional pentru streamuri de intrare *ifstream*, pentru care valoarea implicită este ios::in, şi pentru streamuri de ieşire *ofstream*, pentru care valoarea implicită este ios::out. Pentru streamuri de intrare/ieşire *fstream*, argumentul mod\_de\_deschidere trebuie să aibă una din valorile definite în clasa *ios* :  ios::in este permisa doar citirea dintr-un fişier existent;  ios::out creează un nou fişier, sau dacă există deja, distruge vechiul conţinut;  ios::app deschide un fişier pentru adăugare, pointerul de acces se plasează la sfârșit;  ios::ate deschide un fişier existent pentru citire sau scriere, pointerul de acces se plasează la sfârșit;  ios::trun deschide un fişier şi ştergere vechiul conţinut;  ios::binary specifică fişier de tip binar.  În argumentul mod\_de\_deschidere se pot combina prin operatorul OR (SAU) două sau mai multe din aceste valori definite.  Implicit, fişierele se deschid în mod text. Valoarea ios::binary determină deschiderea în mod binar a fişierului. Orice fişier poate fi deschis în mod text sau mod binar, indiferent de felul în care au fost formatate datele.  Scrierea datelor în fişier se realizează prin intermediul operatorilor de ieşire:  Nume\_file<<flux1<<flux2<<…<<fluxn;  Citirea datelor din fişier se realizează prin intermediul operatorilor de intrare:  Nume\_file >>var1>>var2>>…>>varn;  Valoarea argumentului acces determină tipul de acces la fişier.   * **int** eof (); returnează nenul dacă este sfârșit de fişier; * **int** bad(); returnează nenul dacă s-a produs o eroare; * **int** fail(); returnează nenul dacă operaţia nu a reuşit. * 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 distanţa (în bytes) a pointerului de fişier faţă de începutul fişierului; * seekg(positie, seek\_dir) setează distanţa (în bytes) a pointerului de fişier faţă de poziţia specificată seek\_dir, care poate lua următoarele valori:   ios::beg - început de fişier  ios::cur - poziţie curentă  ios::end - sfârșit de fişier   * tellg(pos) returnează poziţia (în bytes) a pointerului de fişier faţă de începutul fişierului; * write(str, SIZE) inserează SIZE caractere din vectorul str în stream. * tellp() returnează poziţia pointerului de fişier în bytes. | | |

**Probleme pentru rezolvare independentă**

1. Fişierul *date.in* conţine 3 valori separate printr-un singur spaţiu, care reprezintă vârsta, înălţimea şi genul unei persoane. Să se scrie un program care determină greutatea ideală a unei persoane cunoscând înălţimea, vârsta şi genul persoanei (f sau m). Formulele de calcul sunt:

*Gmasculin = 50 + 0.75 \* (inaltime - 150) + (varsta - 20) / 4,*

*Gfeminin = Gmasculin – 10,*

unde înălţimea este exprimată în cm şi vârsta în ani.. La ecran se va afişa greutatea ideală. Toate datele despre persoană (vârsta, înălţimea, genul, greutatea ideală) se vor scrie în fişierul *date.out*.

1. Fişierul *date.in* conţine 3 numere întregi separate printr-un singur spaţiu. Să se elaboreze un program care va afişa aceste numere unul sub altul, afișând în dreptul fiecăruia unul dintre cuvintele PAR sau IMPAR. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
2. Fişierul *date.in* conţine 3 numere întregi distincte separate printr-un singur spaţiu. Să se elaboreze un program care va specifica care dintre numere are valoare maximală, minimală şi care nu este nici minim nici maxim. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*.
3. Fişierul *date.in* conţine 12 numere întregi separate printr-un singur spaţiu care reprezintă temperaturile medii ale lunilor unui an, ca numere întregi. Să se afişeze cu două zecimale media anuală a temperaturilor pozitive şi a celor negative. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out.*
4. Fişierul *date.in* conţine *n* numere întregi separate printr-un singur spaţiu. Primul număr reprezintă numărul de numere din fişier. Datele din fişier reprezintă mediile a n elevi, ca numere reale. Să se afişeze cea mai mare şi cea mai mică medie. Să se verifice dacă sunt corigenţi. Rezultatul va fi afişat la ecran cât şi în fişierul date.out Exemplu : Date de intrare 4 9.50 4.25 9.66 6.33 Date de ieşire max=9.66 min=4.25 1 corigent.
5. Fişierul *date.in* conţine *n* numere întregi separate printr-un singur spaţiu. Primul număr reprezintă numărul de numere din fişier. Să se afişeze numărul minimal, maximal, numărul de elemente pare, impare şi media numerelor cu două zecimale. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
6. Fişierul *date.in* conţine elementele unei matrice. În prima linie sunt scrise numărul de lini şi de coloane separate printr-un spaţiu. Pe următoarele linii sunt scrise elementele matricei. Elaboraţi un program care va afişa la ecran şi va scrie în fişierul *date.out* elementele matricei şi poziţiile pe care se află valoarea maximală.
7. Fişierul *date.in* conţine elementele unei matrice. În prima linie sunt scrise numărul de lini şi de coloane separate printr-un spaţiu. Pe următoarele linii sunt scrise elementele matricei. Elaboraţi un program care va afişa la ecran si va scrie în fişierul *date.out* elementele matricei şi suma de pe fiecare linie.
8. Fişierul *date.in* conţine elementele unei matrice. În prima linie sunt scrise numărul de lini şi de coloane separate printr-un spaţiu. Pe următoarele linii sunt scrise elementele matricei. Elaboraţi un program care va afişa la ecran si va scrie în fişierul *date.out* elementele matricei şi minimul de pe fiecare coloană.
9. Fişierul *date.in* conţine un şir de caractere. Elaboraţi un program care va determina numărul de majuscule din şir. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
10. Fişierul *date.in* conţine un şir de caractere. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va înlocui cifrele din şir cu semnul +. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
11. Fişierul *date.in* conţine un şir de caractere. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va afişa doar cifrele din şir. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
12. Fişierul *date.in* conţine un şir de caractere. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va determina numărul de cifre din şir. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
13. Fişierul *date.in* conţine un şir de caractere. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va înlocui vocalele din şir cu spaţiu. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
14. Fişierul *date.in* conţine un şir de caractere. De la tastatură se citeşte un şir de caractere. Elaboraţi un program care va determina numărul de caractere speciale (paranteze, operatori aritmetici) din şir. Rezultatul va fi afişat la ecran cât şi în fişierul *date.out*
15. În fişierul *numere.in* sunt scrise mai multe numere întregi. Elaboraţi un program prin intermediul căruia se vor adăuga numerele pare din fişierul *numere.in* în fişierul *date.txt*.
16. Elaboraţi un program prin intermediul căruia va fi creat un fişier. Numele fişierului se va citi de la tastatură. În fişier vor fi scrise toate numerele pare mai mici decât n(n<10000), n se citeşte de la tastatură.
17. Elaboraţi un program prin intermediul căruia va fi creat un fişier. Numele fişierului se va citi de la tastatură. În fişier vor fi scrise toate numerele impare mai mici decât n(n<10000), n se citeşte de la tastatură.
18. Fişierul *numere.in* conţine mai multe numere reale (cel puţin 2). Elaboraţi un program prin intermediul căruia în acest fişier se va adăuga media aritmetică a ultimelor 2 numere.
19. Fişierul *numere.in* conţine mai multe numere reale (cel puţin 2). Elaboraţi un program prin intermediul căruia în acest fişier se va adăuga media aritmetică a numerelor din fişier.
20. În fişierul *date.in* este scris un număr întreg pozitiv mai mic decât 1000. Elaboraţi un program care va adăuga în fişier un triunghi:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4  1  12  123  1234 | 4  1234  123  12  1 | 4  1  21  321  4321 |

1. În fişierul *date.in* sunt scrise mai multe cuvinte, fiecare cuvânt este separat printr-un singur spaţiu. Elaboraţi un program prin intermediul căruia se va afişa la ecran conţinutul fişierului, cel mai lung cuvânt şi 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++** |
| *var a : array [1..100] of integer;*  *x,i,j,n,k,l:integer; p:boolean;* | *int a[100],x, i,j,n,k,l.p;* |
| Se cunosc: valoarea lui *n* – numărul de elemente al tabloului, și elementele tabloului *a*. | |
| **Metoda de inserție** | |
| *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;* | ***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;*  *}* |
| **Metoda bulelor** | |
| *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;* | *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++;*  *};*  *};* |
| **Metoda de selecţie** | |
| *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; 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;* | *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];*  *a[l]=k;*  *i++;*  *};* |
| **Sortarea prin distribuire**  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 *a* – datele inițiale și *b* - cel cu rezultate, domeniul fiind de la c la *d*. | |
| *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;* | ***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++;*  *}* |

**Probleme pentru rezolvare independentă**

1. Fişierul *vector.in* conţine n numere întregi separate printr-un singur spaţiu. Primul număr reprezintă numărul de numere din fişier. Elaboraţi un program prin intermediul căruia la ecran se vor afişa numerele sortate crescător şi descrescător, utilizând metoda bulelor, inserţiei şi selecţiei.
2. Fişierul *caractere.in* conţine mai multe caractere. Elaboraţi un program prin intermediul căruia la ecran se vor afişa caracterele sortate crescător şi descrescător, metoda bulelor, inserţiei şi selecţiei.
3. Fişierul *vector.in* conţine n numere întregi separate printr-un singur spaţiu. Primul număr reprezintă numărul de numere din fişier. Elaboraţi un program prin intermediul căruia primele k numere vor fi sortate crescător, iar restul numerelor descrescător, k se citeşte de la tastatură.
4. Fişierul *vector.in* conţine n numere întregi separate printr-un singur spaţiu. Primul număr reprezintă numărul de numere din fişier. Elaboraţi un program prin intermediul căruia primele k numere vor fi sortate descrescător, iar restul numerelor crescător, k se citeşte de la tastatură.
5. Fişierul *cuvinte.in* conţine mai multe cuvinte, separate printr-un singur spaţiu. Elaboraţi un program prin intermediul căruia la ecran se vor afişa cuvintele aranjate descrescător/crescător.
6. Fişierul *matrice.in* conţine elementele unei matrice. În prima linie sunt scrise numerele m şi n – numărul de lini şi de coloane separate printr-un spaţiu. Pe următoarele n linii sunt scrise câte m elemente ale matricei. Elaboraţi un program prin intermediul căruia se vor sorta elementele fiecărei coloane în parte a matricei.
7. Fişierul *matrice.in* conţine elementele unei matrice. În prima linie sunt scrise numărul de lini şi de coloane separate printr-un spaţiu. Pe următoarele linii sunt scrise elementele matricei. Elaboraţi un program prin intermediul căruia se vor sorta elementele fiecărei linii a matricei.

|  |  |
| --- | --- |
| **§6** | **Tipul de date articol** |

1. De la tastatură se citeşte numele şi data naşterii a trei copii. Elaboraţi un program prin intermediul căruia la ecran se va afişa copilul cu vârsta cea mai mică, cea mai mare şi cea mijlocie.
2. De la tastatură se citesc datele despre două variabile de tipul *data* (an, luna, zi). Elaboraţi un program prin intermediul căruia la ecran se va afişa suma şi diferenţa dintre cele două variabile de tipul *data*. La ecran data va fi afişată corect (0≤*zi*≤30, 0≤*luna*≤12)
3. De la tastatură se citesc datele despre două variabile de tipul *timp* (ore, min, sec). Elaboraţi un program prin intermediul căruia la ecran se va afişa suma şi diferenţa dintre cele două variabile de tipul *timp*. La ecran timpul va fi afişată corect (0≤*sec*≤59, 0≤*min*≤59).
4. Elaboraţi un program prin intermediul căruia:

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

a) Dacă coordonatele introduse pot forma un triunghi;

b) În cazul în care triunghiul poate fi format:

-) se va afişa aria şi perimetrul triunghiului;

-) tipul triunghiului: ascuţit, obtuz, drept.

B. De la tastatură se citesc datele despre două cercuri(coordonatele centrului şi raza). Determinaţi:

a) Pentru fiecare cerc aria şi lungimea discului;

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

Formule necesare:

Distanţa dintre 2 puncte A(x1,y1) şi B(x2,y2) 

Aria unui triunghi: , unde

1. Elaboraţi un program prin intermediul căruia se va crea tipul de date fracţie. De la tastatură se citesc datele despre două fracţii, la ecran se va afişa:

a) ce mai mare şi ce mai mică fracţie;

b) suma fracţiilor;

c) diferenţa fracţiilor;

d) produsul fracţiilor;

e) câtul fracţiilor;

Fracţiile se vor afişa doar sub formă de fracţii ireductibile de forma a/b, de exemplu 3/4.

1. Pe o suprafaţă plană sunt mai multe puncte, date prin coordonatele lor. Elaboraţi un program prin intermediul căruia se vor sorta crescător în dependenţă de distanţa acestora de la centru axei de coordonate a sistemului cartezian.

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

**Date de ieşire**. La ecran se va afişa punctele sortate, fiecare coordonată de punct fiind urmată de distanţa acestuia de la centru.

1. Prin intermediul tipului de date **articol**, creaţi tipul de date telefon cu câmpurile: marca, culoare, anul fabricării, preţul. În baza acestui tip elaboraţi un program prin intermediul căruia se vor gestiona datele despre telefoanele mobile (minim 10 telefoane). Citirea datelor se va efectua din fişier. Programul va conţine un meniu cu următoarele opţiuni:

a) Afişarea tuturor telefoanelor;

b) Afişarea telefoanelor cu cel mai mare an;

c) Afişarea celor mai scumpe telefoane;

d) Căutarea telefoanelor după marcă;

e) Căutarea telefoanelor după preţ, cu indicarea preţului minim şi maxim;

f) Sortarea telefoanelor descrescător după anul fabricării;

g) Sortarea telefoanelor crescător după marcă.

|  |  |
| --- | --- |
| **§7** | **Recapitulare** |

1. Se consideră două tablouri ***A[1..N]***şi ***B[1..N]***de numere reale, ***N ≤ 100***. Scrieţi un program care numără cazurile cu următoarea proprietate:

***A[i]<B[i]; A[i]=B[i];*** A[i]>B[i].

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

*Ieşire:* La ecran se vor afişa trei rânduri, indicând proprietatea şi numărul de situaţii respective.

1. Se consideră tabloul ***A[1..N]*** de numere întregi, ***N ≤ 100***. Scrieţi un program care calculează media aritmetică a numerelor de pe locurile pare şi media aritmetică a numerelor de pe locurile impare.

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

*Ieşire:* La ecran se vor afişa două valori.

1. Se consideră tabloul ***A[1..N]***de numere întregi, ***N ≤ 100***. Scrieţi un program care calculează suma elementelor mai mici ca elementul maxim.

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

*Ieşire:* La ecran se va afişa suma cerută.

1. Se consideră tabloul ***X[1..N]*** de numere întregi, ***N ≤ 100***. Scrieţi un program care determină lungimea celei mai lungi secvenţe de elemente consecutive ordonate crescător.

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

*Ieşire:* La ecran se va afişa mărimea celei mai lungi secvenţe de elemente ordonate crescător şi poziţiile începutului şi sfârșitului ei.

1. Se consideră tabloul ***A[1..N]***de numere întregi, ***N ≤ 100***. Scrieţi un program care efectuează deplasarea spre stânga cu o poziţie a elementelor ***A[2], A[3],..., A[n].*** Primul element va fi înscris pe ultima poziţie.

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

*Ieşire:* La ecran se vor afişa elementele tabloului modificat.

1. Se consideră tabloul ***A[1..N]*** de numere întregi, ***N ≤ 100***. Scrieţi un program care efectuează deplasarea elementelor spre stânga cu ***M*** poziţii. Primele ***M*** elemente vor fi înscrise pe ultimele ***M*** poziţii.

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

*Ieşire:* La ecran se vor afişa elementele tabloului modificat.

1. Se consideră tabloul ***X[1.. 10]*** de numere întregi egale numai cu *0, 1* sau *2*. Scrieţi 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.

*Ieşire:* La ecran se va afişa elementele tabloului modificat.

1. Se consideră tabloul ***A[1..2\*N]*** de numere naturale, unde ***N*** elemente sunt pare, iar ***N*** elemente sunt impare, ***N ≤ 50***. Scrieţi 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 obţine: 5 4 7 6 9 8

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

*Ieşire:* La ecran se vor afişa elementele tabloului modificat.

1. Se consideră tabloul ***X[ 1 .. N]*** de numere reale, ***N ≤ 100***. Scrieţi 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*** şi elementele tabloului se citesc de la tastatură.

*Ieşire:*  *E*lementele tabloului ***Y*** se vor afişa la ecran.

1. Se consideră tabloul ***T1..N, 1..M***cu numere reale**, *N ≤ 20*** *şi* ***M ≤ 20***. Scrieţi un program care interschimbă cu locurile elementul maxim de pe fiecare coloană cu elementul din prima linie a aceleiaşi coloane.

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

*Ieşire:* La ecran se vor afişa elementele tabloului modificat

1. Se consideră două tablouri liniare ***A1..N*** şi ***B1..N, N ≤ 100***. Scrieţi un program care modifică elementele acestor tablouri conform regulii: ***Ai:= Ai + Bi şi Bi:= Ai\* Bi.*** De exemplu, dacă ***Ai= 7 şi Bi = 3*** atunci după modificare obţinem valorile lui ***Ai = 10*** şi ***Bi = 21.***

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

*Ieşire:* La ecran se vor afişa elementele tablourilor modificate.

1. Se consideră tabloul ***X[1..N]*** de numere întregi nenule, ***N ≤ 100***. Scrieţi 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*** şi elementele tabloului se citesc de la tastatură.

*Ieşire:* La ecran se va afişa numărul de alternări de semne.

|  |  |
| --- | --- |
| **Capitolul 3** | **SUBPROGRAME** |

|  |  |
| --- | --- |
| **§1** | **Parametri formali de tip valoare** |

1. De la tastatură se citeşte un număr întreg n, n<100000. Elaboraţi un program cu funcţii prin intermediul căruia se va determina:
2. Numărul de cifre;
3. Numărul de cifre pare;
4. Numărul de cifre impare;
5. Suma cifrelor;
6. Cifra maximă;
7. Cifra minimă;
8. Media aritmetică a cifrelor;
9. Cifrele care se repetă de cel puţin două ori;
10. Cifrele numărului separate printr-un sigur spaţiu;
11. Divizorii numărului;
12. Inversul numărului;
13. Mesajul PRIM, dacă numărul este număr prim;
14. Cel mai mare număr posibil creat din aceste cifre.
15. De la tastatură se citesc două numere întregi. Elaboraţi un program cu funcţii prin intermediul căruia se va determina:
16. Suma numerelor;
17. Produsul numerelor;
18. Media aritmetică a numerelor;
19. Cel mai mare divizor comun;
20. Cel mai mic multiplu comun;
21. Numărul minim;
22. Numărul maxim;
23. Suma numerelor în formatul *a+b=c*, dacă *a* şi *b* reprezintă numerele citite;
24. Produsul numerelor în formatul *a\*b=c*, dacă *a* şi *b* reprezintă numerele citite;
25. Toţi divizorii comuni;
26. Cinci multipli comuni;
27. Cifrele care se conţin în ambele numere;
28. Cifrele care sunt în primul număr şi nu sunt în al doilea număr;
29. Va afişa mesajul PRIETENE, dacă numerele sunt prietene. Două numere se numesc prietene, dacă numărul de divizori este acelaşi.
30. De la tastatură se citesc trei numere întregi. Elaboraţi un program cu funcţii prin intermediul căruia:
31. Se va determina cel mai mare divizor comun al numerelor;
32. Se va determina cel mai mic multiplu comun al numerelor;
33. Se va determina valoarea maximă;
34. Se va determina valoarea minimă;
35. Se vor afişa toţi divizorii comuni;
36. Se vor cei mai mici trei multipli comuni.
37. 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;

1. Se va afişa soluţiile reale ale ecuaţiei *ax2+bx+c=0*, dacă *a*, *b*, *c* reprezintă coeficienții ecuaţiei.

|  |  |
| --- | --- |
| **§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:
2. Suma numerelor;
3. Produsul numerelor;
4. Media aritmetică a numerelor;
5. Cel mai mare divizor comun;
6. Cel mai mic multiplu comun;
7. Numărul minim;
8. Numărul maxim.
9. Se consideră tipul de date *timp (ora:min:sec)*. Elaboraţi un program prin intermediul căruia se va crea câte o procedură pentru:
10. Citirea timpului de la tastatură;
11. Afişarea timpului la ecran, în formatul *ora:min:sec*, dacă 0≤*sec*≤59 şi 0≤*min*≤59.
12. Determinarea sumei dintre cele două variabile de tipul timp;
13. Determinarea diferenţei dintre cele două variabile de tipul timp;
14. Determinarea valorii maximale;
15. Determinarea valorii minimale;
16. Se consideră tipul de date *data (an:luna:zi)*. Elaboraţi un program prin intermediul căruia se va crea câte o procedură pentru:
17. Citirea datei de la tastatură;
18. Afişarea datei la ecran, în formatul *an:luna:zi*, dacă 1≤*zi*≤30 şi 1≤*luna*≤12.
19. Determinarea sumei dintre cele două variabile de tipul data;
20. Determinarea diferenţei dintre cele două variabile de tipul data;
21. Determinarea valorii maximale;
22. Determinarea valorii minimale;
23. Se consideră tipul de date *polinom (ax2+bx+c)*. Elaboraţi un program prin intermediul căruia se va crea câte o procedură pentru:
24. Citirea coeficienților polinomului de la tastatură;
25. Afişarea polinomului la ecran, în formatul *ax2+bx+c*, dacă a, b, c reprezintă coeficienții polinomului.
26. Determinarea sumei dintre cele două polinoame;
27. Determinarea diferenţei dintre cele două polinoame;
28. Determinarea polinomului maxim;
29. Determinarea polinomului minim;
30. Determinarea soluţiilor reale ale polinomului.
31. 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:
32. Se va citi de la tastatură lungimile laturilor;
33. Se va determina dacă valorile a,b şi c au fost introduse corect(pot forma un triunghi), în caz contrar se va repeta operaţia de citire, până când valorile nu vor fi introduse corect;
34. Se va determina aria şi perimetrul triunghiului.
35. Se va afişa tipul triunghiului (ascuţit, drept, obtuz).

|  |  |
| --- | --- |
| **§3** | **Tablouri şi funcţii** |

1. Se consideră un tablou unidimensional. Elaboraţi un program cu funcţii/proceduri prin intermediul cărora:
2. De la tastatură se vor citi elementele tabloului;
3. Se vor afişa elementele tabloului la ecran;
4. Se vor sorta elementele crescător;
5. Se vor inversa elementele tabloului;
6. Va fi determinat elementul maximal;
7. Va fi determinat elementul minimal;
8. Se va determina numărul de elemente maximale;
9. Se va determina numărul de elemente minimale.
10. Se consideră două tablouri cu numere întregi, a şi b. Elaboraţi un program utilizarea funcţiilor/procedurilor prin intermediul cărora:
11. De la tastatură se citesc elementele tablourilor;
12. Se afişează la ecran elementele tablourilor;
13. Va determina suma elementelor fiecărui tablou;
14. Va determina diferenţa elementelor fiecărui tablou;
15. Va crea un nou tablou care va conţine toate elementele tabloului a şi b;
16. Va crea un nou tablou format doar din elementele comune ambelor tablouri.
17. Se consideră un tablou bidimensional cu numere întregi. Elaboraţi un program cu funcţii/proceduri prin intermediul cărora:
18. De la tastatură se vor citi elementele tabloului;
19. Se vor afişa elementele tabloului la ecran;
20. Va fi determinat elementul maximal;
21. Va fi determinat elementul minimal;
22. Se va determina numărul de elemente maximale;
23. Se va determina numărul de elemente minimale;
24. Se va afişa sumele pe linii;
25. Se va afişa sumele pe coloane;
26. Se vor sorta elementele crescător pe linii;
27. Va determina suma elementelor de pe *perimetrul* tabloului.
28. Se consideră un şir de caractere. Elaboraţi un program cu funcţii/proceduri, prin intermediul căruia:
29. Va determina dacă şirul este palindrom;
30. Va afişa doar vocalele din şir;
31. Va afişa doar cifrele din şir;
32. Va determina numărul de cuvinte din şir;
33. Va sorta crescător elementele şirului;
34. Va inversa şirul;
35. Va determina cuvântul maxim;
36. Va determina cuvântul de lungime maximă;
37. Va inversa fiecare cuvânt din şir.
38. Elaboraţi un program prin intermediul căruia vor fi create proceduri pentru conversia numerelor în şir de caractere şi reciproc:
39. Procedură pentru transformarea şirului în număr întreg, dacă şirul conţine doar cifre;
40. Procedură pentru transformarea unui număr întreg în şir;
41. Procedură pentru transformarea şirului în număr real, dacă şirul conţine doar cifre şi caracterul *punct*;
42. Procedură pentru transformarea unui real în şir.

|  |  |
| --- | --- |
| **§4** | **Subprograme recursive** |

1. De la tastatură se citeşte un număr întreg. Elaboraţi un subprogram recursiv, prin intermediul căruia se va determina valoarea expresiilor:
2. 1+3+5+…+2n-1;
3. 2+4+6+…+2n;
4. 2+5+8+…+3n-1;
5. 1+5+9+…+4n-3;
6. ;
7. ;
8. .
9. Se consideră un tablou unidimensional cu numere întregi. Elaboraţi un subprogram recursiv prin intermediul căruia:
10. De la tastatură se citesc elementele acestui tablou unidimensional;
11. Se vor afişa elementele tabloului în ordinea citirii lor;
12. Se vor afişa elementele tabloului în ordine inversă a citirii lor;
13. Va determina suma elementelor;
14. Va determina suma elementelor pare;
15. Va determina suma elementelor impare;
16. Va afişa elementele pare;
17. Va afişa elementele impare;
18. Va verifica existenţa numărului k în tablou (k se introduce de la tastatură);
19. Va verifica existenţa numerelor pare în tablou.
20. Elaboraţi un subprogram recursiv prin intermediul căruia:
21. Se va determina produsul dintre două numere întregi prin adunări repetate;
22. Se va determina câtul dintre două numere întregi prin scăderi repetate;
23. Se va determina restul împărţirii dintre două numere întregi prin scăderi repetate;
24. Elaboraţi un subprogram recursiv, prin intermediul căruia se va determina dacă un număr întreg, este sau nu număr prim.

|  |  |
| --- | --- |
| **§5** | **Recapitulare** |

1. Se consideră tabloul ***A[1..N,1..N]*** de numere întregi, unde ***N ≤ 20***. Scrieţi un program care va aduna la fiecare element al tabloului valoarea elementului maxim de pe diagonala principală.

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

*Ieşire:* La ecran se vor afişa elementele tabloului modificat.

1. Se consideră tabloul ***A[1..N,1..N]*** de numere întregi, unde ***N ≤20***. Scrieţi un program care calculează suma elementelor care nu se află pe diagonala secundară.

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

*Ieşire:* La ecran se va afişa suma calculată.

1. Se consideră un tablou pătrat (numărul de linii este egal cu numărului de coloane) elementele căruia sunt numere întregi. Scrieţi un program care modifică tabloul astfel că toate elementele acesteia se măresc cu valoarea elementului situat pe diagonala secundară şi în aceeaşi linie.

*Intrare: F*işierul text ***TABEL.IN***organizat astfel: prima linie al fişierului conţine un număr pozitiv întreg ***1< N ≤ 10;*** celelalte ***N***  linii conţin câte ***N*** valori numerice separate prin spaţii

*Ieşire:* Fişierul text ***TABEL.OUT*** conţine ***N*** linii a câte ***N*** valori separate prin spaţii. *De exemplu:*

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

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

*Intrare: F*işierul text ***TABEL.IN***organizat astfel: prima linie al fişierului conţine un număr pozitiv întreg ***1< N ≤ 10;*** celelalte ***N***  linii conţin câte ***N*** valori numerice separate prin spaţii

*Ieşire:* Fişierul text ***TABEL.OUT*** conţine ***N*** linii a câte ***N*** valori separate prin spaţii

*De exemplu:*

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

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

*Intrare: F*işierul text ***TABEL.IN***organizat astfel: prima linie al fişierului conţine un număr pozitiv întreg ***1< N ≤ 10;*** celelalte ***N***  linii conţin câte ***N*** valori numerice separate prin spaţii

*Ieşire:* Fişierul text ***TABEL.OUT*** conţine ***N*** linii a câte ***N*** valori separate prin spaţii. *De exemplu:*

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

1. Se consideră tabloul liniar ***T1..N*** de tip întreg, ***N≤100***. Scrieţi un program care înscrie numerele nenule la începutul tabloului (păstrând ordinea lor) iar zerourile le plasează la finele tabloului. *Restricţie*: se interzice utilizarea altor tablouri!

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

*Ieşire:* La ecran se vor afişa elementele tabloului modificat

1. Fişierul cu acces direct ***DATE.DAT*** conţine 3\*n numere reale (n triplete). Scrieţi un program care modifică fişierul astfel: fiecare al doilea element din triplete consecutive se înlocuieşte cu media aritmetică a vecinilor lui. De exemplu, dacă ***DATE.DAT*** conţinea *4 7 6 8 2 10 1 5 7* atunci după îndeplinirea programului va conţine *4 5 6 8 9 10 1 4 7*.

*Intrare*: Fişierul ***DATE.DAT***

*Ieşire*: Fişierul ***DATE.DAT*** modificat.

1. Fişierul text **ELEVI.IN** conţine datele referitoare la examenele de admitere a cel mult 200 concurenţi. Fiecărui elev îi corespund câte două linii din acest fişier, pe prima aflându-se numele şi prenumele său, iar pe a doua media obţinută la examene. Scrieţi un program care scrie în fişierul **ELEVI.OUT** lista elevilor admişi (care au media nu mai mică decât 5) ordonată descrescător după medii.

*Intrare:* Fişierul text **ELEVI.IN**

*Ieşire:* Fişierul text **ELEVI.OUT**, ce conţine pe fiecare linie numele şi prenumele elevilor admişi, separate printr-un spaţiu.

1. În fişierul text ***ELEVI.INT*** se conţin date despre toţi elevii absolvenţi din Republica Moldova. În fiecare rând sunt înscrise datele despre o singură persoană astfel: nume, prenume, sex şi înălţimea, despărţite printr-un spaţiu. Din aceste date se va crea o listă simplu înlănţuită ordonată alfabetic, după nume. Parcurgeţi lista în scopul calculării şi afişării la ecran a înălţimii medii a fetelor şi a băieţilor.

*Intrare:* Fişierul text ***ELEVI.INT***, unde sexul este indicat prin ***f*** pentru fete şi ***b*** pentru băieţi.

*Ieşire:* La ecran se vor afişa mesajul respectiv şi mediile cerute.

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

Scrieţi un program care citeşte din fişierul text ***DATE.IN*** câte un număr de 4 cifre din fiecare linie, şi creează alt fişier text ***DATE.OUT*** unde înscrie numerele „bine ordonate”. La ecran se va afişa numărul lor.

*Intrare:* Fişierul text ***DATE.IN***

*Ieşire:* Fişierul text ***DATE.OUT*** La ecran se va afişa numărul de numere „bine ordonate”.

1. Se consideră un fişier text, ***INTRARE.TXT***, care conţine cuvinte separate prin câte un singur spaţiu. Să se creeze un fişier text, ***IESIRE.TXT***, care conţine cuvintele textului în ordine alfabetică şi numărul lor de apariţii. Cuvintele textului apar o singură dată în fişierul de ieşire.

*Intrare:* Fişierul text ***INTRARE.TXT***

*Ieşire:* Fişierul text ***IESIRE.TXT***

*Exemplu:*

Pentru fişierul ***INTRARE.TXT***:

A fost odata ca-n povesti

A fost ca niciodata

Din rude mari imparatesti

O prea frumoasa fata

fişierul ***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

1. Se consideră un fişier text ***TEST.IN*** alcătuit din mai multe linii. Scrieţi un program care să afişeze pe ecran toate literele alfabetului englez, folosite în acest fişier, împreună cu codul lor ASCII şi frecvenţa lor de apariţie. Afişaţi, de asemenea, codurile ASCII ale caracterelor care apar în text cel mai frecvent.

*Intrare:* Fişierul text ***TEST.IN.***

*Ieşire:* La ecran se vor afişa datele cerute.

De exemplu, pentru fişierul de intrare:

aBc bAaccaa

bc ac

pe ecran se va afişa:

|  |  |  |
| --- | --- | --- |
| 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

1. Se consideră fişierul text ***CUVINTE.TXT*** ce conţine mai multe cuvinte formate din litere minuscule ale alfabetului latin. Scrieţi un program care înscrie în fişierul ***CUVINTE.OUT***, în ordinea alfabetică literele distincte din componenţa fişierului ***CUVINTE.TXT*** şi numărul respectiv de apariţii al fiecărei litere. Rezultatul se înscrie în fiecare rând câte o literă şi numărul de apariţii, separate printr-un spaţiu.
2. Se cunoaşte că fişierul text ***INPUT.TXT*** este alcătuit din numărul par de linii şi conţine în linii impare numele elevilor, dar în cele pare – notele acestora la o lucrare de control. Numărul maxim al elevilor în clasă – 40. Scrieţi un program care citeşte fişierul dat şi creează un fişier text ***SORT.TXT*** care conţine în fiecare rând numele elevului şi nota în aşa 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. | *New(var p:POINTER);* |
| Eliberarea memoriei alocate în heap pentru o variabila referință legată. | *Dispose(var p:POINTER);* |
| Lungimea celui mai mare bloc continuu existent în heap | *MaxAvail:LONGINT;* |
| Lungimea totala a spațiului de memorie disponibil în heap. | *MemAvail:LONGINT;* |
| Alocarea memoriei în heap pentru o variabila dinamica de mărime l si încărcarea adresei ei în argumentul p. | *GetMem(VAR p:POINTER; l:WORD);* |
| Eliberarea memoriei alocate în heap pentru o variabila dinamica de mărime l si de adresa memorata în p. | *FreeMem(VAR p:POINTER; l:WORD);* |
| Memorarea adresei vârfului heap-ului (HeapPtr) în argumentul p. | *Mark(VAR p:POINTER);* |
| Eliberarea memoriei alocate variabilelor dinamice prin apelul precedent al procedurii Mark (depune p în HeapPtr). | *Release(VAR p:POINTER);* |

1. Prin intermediul structurilor dinamice de date, creaţi tipul de date telefon cu câmpurile: marca, culoare, anul fabricării, preţul. În baza acestui tip elaboraţi un program prin intermediul căruia se va gestiona datele despre telefoanele mobile (minim 10 telefoane). Citirea datelor se va efectua din fişier. Programul va conţine un meniu cu următoarele opţiuni:
2. Afişarea tuturor telefoanelor;
3. Afişarea celor mai noi telefoane;
4. Afişarea celor mai scumpe telefoane;
5. Căutarea telefoanelor după marcă;
6. Căutarea telefoanelor după preţ, cu indicarea preţului minim şi maxim;
7. Sortarea telefoanelor descrescător după anul fabricării;
8. Sortarea telefoanelor crescător după marcă.
9. Inserare datelor despre un telefon nou;
10. Excluderea unui telefon.
11. De la tastatură se citesc cuvinte. Să se creeze o listă simplu înlănţuită ordonată alfabetic, care conţine în celule cuvintele distincte şi frecvenţa lor de apariţie. Se va afişa conţinutul listei în ordine alfabetică
12. De la tastatură se citesc mai multe numere întregi. Să se creeze o listă ordonată crescător.
13. De la tastatură se citeşte numărul n şi 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.
14. Se consideră un depou de locomotive cu o singură intrare şi 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 ieşire din depou a unei locomotive şi de afişare a locomotivelor din depou.
15. De la tastatură se citesc cuvinte (şiruri de caractere). Să se scrie un program care creează un arbore de căutare, care conţine în noduri cuvintele şi frecvenţa lor de apariţie. Să se afişeze apoi cuvintele în ordine lexicografică crescătoare şi frecvenţa lor de apariţie.
16. Informaţiile pentru medicamentele unei farmacii sunt: nume medicament, preţ, cantitate, data primirii, data expirării. Evidenţa medicamentelor se ţine cu un program care are drept structură de date un arbore de căutare după nume medicament. Să se scrie programul care execută următoarele operaţii:
17. creează arborele de căutare;
18. caută un nod după câmpul nume medicament şi actualizează câmpurile de informaţie;
19. tipăreşte medicamentele în ordine lexicografică;
20. elimină un nod identificat prin nume medicament;
21. creează un arbore de căutare cu medicamentele care au data de expirare mai veche decât o dată specificată de la terminal.
22. Se consideră o listă unidirecţională câmpul informaţional al fiecărei celule conţine un număr real. Scrieţi declaraţiile necesare şi fragmentul de program care ar adăuga între oricare două celule vecine a listei iniţiale celule noi câmpul informaţional al cărora va fi media aritmetică a numerelor din celulele vecine.

*Ieşire*: La ecran se afişează valorile câmpurilor informaţionale ale celulelor listei modificate.

|  |  |
| --- | --- |
| **Capitolul 5** | **TEHNICI DE PROGRAMARE** |

|  |  |
| --- | --- |
| **§1** | **Metoda trierii** |

1. Se consideră numerele naturale din mulţimea {1, 2, 3, ..., n}, n<100000, n se citeşte din fişier. Să se determine toate elementele acestei mulţimi, pentru care suma cifrelor este un număr impar, divizibil cu trei;
2. Fişierul date.in conţine mai multe caractere, să se afişeze cea mai lungă secvenţă de caractere identice. Dacă sunt mai multe secvenţe, se va afişa prima. Exemplu: text: aabsssdadaaab se va afişa sss.
3. Se consideră numerele naturale din mulţimea {1, 2, 3, ..., n}, n<100000, n se citeşte din fişier. Să se determine toate elementele acestei mulţimi, pentru care suma cifrelor este divizibilă cu *a* şi cu *b*, a şi b se citesc din fişier.
4. Se consideră mulţimea ***P={P1,P2,…,Pn}*** formată din ***n*** puncte ***(2≤n≤30)*** pe un plan euclidian. Fişierul *date.in* conţine 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 ***xj , yj*** . Elaboraţi un program care afişează la ecran coordonatele punctelor ***Pa*** , ***Pb*** distanţa dintre care este maximă. Distanţa dintre punctele ***Pj*** ,***Pm*** se calculează cu ajutorul formulei:

******

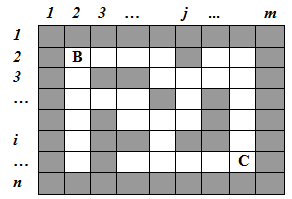
1. Pe o suprafaţă plană sunt mai multe puncte, date prin coordonatele lor. Fişierul *puncte.in*, care pe prima linie conţine numărul de puncte din plan, iar pe următoarele linii sunt scrise coordonatele punctelor. Elaborați un program prin intermediul căruia se determina numărul maximal de triunghiuri care pot fi formate din aceste puncte cu condiţia că:
2. Triunghiurile formate nu se intersectează;
3. Triunghiurile formate sunt înscrise;
4. Triunghiurile formate au un vârf comun.

La ecran se va afişa numărul de triunghiuri.

|  |  |
| --- | --- |
| **§2** | **Metoda Greedy** |

1. **Problema spectacolelor** Într-o sală într-o zi trebuie planificate n spectacole. Pentru fiecare spectacol se cunoaşte intervalul în care se desfăşoară: (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 dispoziţie n obiecte şi cunoaşte pentru fiecare obiect greutatea şi câștigul care se obţine în urma transportului său la destinaţie. Se cere să se precizeze ce obiecte trebuie să transporte persoana în aşa fel încât câștigul sa fie maxim. Persoana are posibilitatea să taie obiectele
3. **Problema bancomatelor** Scrieţi un program, care afişează 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 şi 500 de lei. Numărul de bancnote de fiecare valoare se citeşte din fişierul text ***BANI.IN***, care conţine 7 rânduri, în fiecare din care sunt indicate numărul de bancnote respectiv de 1, 5, 10, 50, 100, 200 şi 500 de lei.
4. **Suma componentelor prime** Fie *a* o variabilă indexată, ale cărei componente A*(1) , A(2),…, 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 consideraţie o singură dată.
5. La un concurs de automobilism de la linia de start până la final sunt plasate n staţii de benzină (la diferite distanţe). Având rezervorul plin, maşina unui concurent poate parcurge cel mult o distanţă d (în kilometri). Concurentul doreşte să se oprească de cât mai puţine ori şi desigur să parcurgă întreg drumul de la punctul de start la cel final. Descrieţi o metodă eficientă pe care trebuie să o aplice concurentul şi arătaţi că strategia respectivă conduce la o soluţie optimă.
6. Pe o bandă magnetică sunt *n* programe. Pentru fiecare program i (1 i n) de lungime li se cunoaşte probabilitatea pi cu care poate fi apelat (cerut), p1 + p2 + ... + pn = 1. Pentru a citi un program trebuie să citim banda de la început (secvenţial). În ce ordine să memorăm programele pentru a minimiza timpul mediu de citire a unui program oarecare?

|  |  |
| --- | --- |
| **§3** | **Metoda Reluării** |



1. **Labirintul** Se consideră planul unui labirint desenat pe o foaie de hârtie liniate în pătrăţele. Pătrăţelele haşurate reprezintă obstacolele, iar cele nehaşurate - camerele şi coridoarele labirintului.

În memoria calculatorului planul labirintului, este redat prin matricea ***A=||aij|| n\*m  1≤n, m≤30,***  unde ***aij***= 1, dacă pătrăţelul este haşurat, a***ij***=0, în caz contrar. Călătorul se poate deplasa dintr-un pătrăţel nehaşurat numai atunci când ele au o latură comună. Elaboraţi un program care găseşte, dacă există, un drum din pătrăţelul ***B*** în pătrăţelul ***C***

1. Elaboraţi un program care afişează la ecran toate modurile de a descompune un număr natural în sumă de ***k*** numere naturale distincte. E*xemplu:* pentru n=9 şi k=3 soluţiile sunt: 1+2+6,2+3+4,1+3+5 3.
2. Se consideră n (n<50) săculeţe, numerotate respectiv: 1, 2, 3,…, *n*. Săculeţul i conţine mi monede de aceiaşi valoare Vi. Elaboraţi un program care afişează la ecran modul de plată a unei sume p cu monedele din săculeţe.
3. Dintr-un fişier se citesc numele şi notele medii a x clase de elevii. Fiecare clasă este formată din minim a şi maxim b studenţi (2<a<b<30). Studenţii necesită a fi grupaţi în grupuri de câte x studenţi, din fiecare grupă este selectat câte un student, astfel încât media notelor grupului să fie egală cu 8. Afişaţi lista tuturor grupelor, care pot fi formate, enumerând numele elevilor din fiecare grupă.
4. Se dau N (*N≤50.000)* dominouri. Să se determine o modalitate de construire al unui şir, care să conţină toate dominourile, respectând regula jocului domino. Aceasta regula prevede ca numerele înscrise pe feţele corespunzătoare a două dominouri consecutive trebuie să fie egale. Dominourile pot fi alese în ordine oarecare şi 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 cate doua numere separate printr-un singur spațiu, reprezentând cele doua numere înscrise pe dominoul corespunzător.

**Date de ieşire**. Pe prima linie a fişierul *domino.out* va trebuie să afişaţi *1*, dacă există soluție şi *0* dacă nu. Dacă există soluţie vor urma exact *N* linii, care descriu şirul construit. Fiecare număr de ordine va fi urmat de un spaţiu şi de un număr, care poate fi *0* sau *1* şi va fi egal cu *1* în cazul în care dominoul corespunzător a fost rotit.

|  |  |
| --- | --- |
| **§4** | **Desparte şi stăpânește** |

1. Utilizând metoda desparte şi stăpânește
2. Să se determine produsul a n numere întregi;
3. Să se determine maximul (minimul) a n numere întregi;
4. Să se determine cel mai mare divizor comun a *n* valori dintr-un vector;
5. Să se caute o valoare într-un vector. Dacă se găsește se va afișa poziţia pe care s-a găsit, altfel se va afişa mesajul Nu exista asa valoare;
6. Să se caute o valoare într-un vector ordonat crescător;
7. Să se numere câte valori sunt egale cu x dintr-un şir de numere întregi citite.
8. Considerăm un şir de *n* numere întregi, ordonat crescător şi un număr întreg *x*. Scrieţi un program care împarte şirul dat în două subşiruri în aşa fel încât toate elementele primului subşir să fie mai mici sau cel mult egale cu *x*, iar toate elementele celui de-al doilea subşir sa fie strict mai mari decât *x*.
9. **Turnurile din Hanoi**. Se dau trei tije, a, b şi c. Pe tija a se află *n* discuri de dimensiuni diferite, ordonate în ordinea diametrelor (discul cel mai mare la bază). Se doreşte mutarea tuturor discurilor de pe tija *a* pe tija *b*, utilizând tija intermediară c, cu condiţia că un disc cu diametrul mai mare să nu fie pus pe vreo tijă peste un disc cu dimensiune mai mică.
10. Intr-o placă dreptunghiulară cu dimensiunile N pe L (coordonatele dreptunghiului) sunt făcute K găuri. Coordonatele găurilor sunt numere întregi. Fişierul *date.in* conţine în prima linie numărul n, care reprezintă numărul de găuri, pe liniile următoare se regăsesc coordonatele găurilor. Scrieţi un program, care determină cel mai mic dreptunghi(cu o suprafaţă mai mare ca 0) fără găuri din placă.
11. Cunoscându-se numărul *n* al elevilor unei clase, precum şi mediile generale ale celor *n* elevi la finele unui an şcolar, realizaţi un program care, folosind un algoritm Divide et Impera, testează dacă în clasa respectivă exista doi elevi cu aceeaşi medie generală. Programul va tipări pe ecran mesajul “da” sau “nu”, în funcţie de situaţie.
12. Intr-o placă dreptunghiulară cu dimensiunile N pe L sunt n puncte (2≤n≤30). Fişierul *date.in* conţine în prima linie numărul n, care reprezintă numărul de puncte din plan pe liniile următoare se regăsesc coordonatele punctelor. Elaboraţi un program care afişează la ecran coordonatele punctelor Pa, Pb distanţa dintre care este minimă. Distanţa dintre punctele Pj , Pm se calculează cu ajutorul formulei:

|  |  |
| --- | --- |
| **§5** | **Recapitulare** |

1. Se consideră tabloul bidimensional T[1:5,1:6]. Elementele lui pot avea valori 0 sau 1. Determinaţi numărul „insuliţelor” 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 depăşeşte pe a celor peste care s-a scurs vopseaua. Curgerea vopselei are loc pe direcţiile orizontală şi verticală. Determinaţi toate elementele peste care s-a prelins vopseaua.
3. Se dau 1000 de monede, dintre care una este mai uşoară decât celelalte, care sunt identice. Determinaţi moneda falsă printr-un număr minim de cântăriri cu o balanţă fără greutăţi marcate.
4. Pe un câmp dreptunghiular se află diferite obiective militare de formă dreptunghiulară. Lăţimea lor este egală cu o unitate, iar lungimea poate fi diferită, dar constituie numere întregi de unităţi de măsură. Determinaţi numărul de obiective de fiecare tip.
5. Un tablou bidimensional are valorile componentelor numere întregi pozitive. Determinaţi patru elemente ce ar reprezenta vârfurile unui pătrat, astfel încât suma acestor elemente să fie maximală.
6. Se dă mulţimea A={1, 2, ..., n}. Să se determine toate permutările acestei mulţimi.
7. Se dă mulţimea A={1, 2, ..., n}. Să se determine toate combinările acestei mulţimi luate câte k elemente.
8. Se dă mulţimea A={1, 2, ..., n}. Să se determine toate partiţiile acestei mulţimi.
9. Un copil ştie doar adunarea cu 1 şi înmulţirea cu 2. Ajutaţi-l, pornind de la 1, să obţină numărul 100 printr-un număr minim de operaţii respective.
10. Se dă un tablou liniar de numere întregi ordonate crescător. Determinaţi dacă o valoare dată m se regăseşte printre componentele tabloului printr-un număr minim de comparări.
11. Se consideră două tablouri liniare de numere întregi sortate crescător. Obţineţi în baza lor al treilea tablou astfel încât el să se obţină sortat la completare.
12. Un călător doreşte să viziteze n oraşe, astfel încât să nu treacă de două ori prin acelaşi oraş, revenind în oraşul de unde a pornit, marcat cu numărul 1. Cunoscând legăturile existente între oraşe, se cere să se tipărească toate drumurile posibile pe care le poate efectua călătorul.
13. Determinaţi toate modalităţile de a plăti o sumă anumită de bani S cu bancnote având valorile nominalelor A1, A2, ..., An. Numărul bancnotelor de fiecare nominal este suficient de mare.
14. Avem la dispoziţie 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.

1. Fie un număr natural N. Determinaţi toate descompunerile sale în sumă de numere prime.
2. Fie N piese. Se cunoaşte timpul necesar pentru prelucrarea fiecărei piese la fiecare dintre două strunguri. Determinaţi în ce ordine trebuie prelucrate piesele la cele două strunguri, ştiind 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.
3. Din fişierul secv.in se citesc un număr natural n şi o matrice pătratică de dimensiune nxn, conținând litere din alfabet. De la tastatură se citeşte un cuvânt. Verificaţi dacă respectivul cuvânt se conţine 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 afişa numărul liniei şi coloanei de unde începe cuvântul şi 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 |

1. Aparatele pentru perforarea tichetelor de călătorie în transportul public folosesc 9 puncte de perforare, dispuse în formă de matrice 3x3. Să se genereze toate modalităţile în care pot fi perforate biletele.
2. Se consideră planul unui labirint, redat prin matricea [A1:n, 1:m], elementele căreia au valori 1, dacă poziţia respectivă reprezintă obstacol, sau 0 – în caz contrar (1<=n, m<=30). Pe plan sunt date două puncte B şi C, ce nu reprezintă obstacole. Elaboraţi un program ce determină dacă există un drum din punctul B până în punctul C. Deplasarea se face doar pe direcţie orizontală sau verticală.
3. Se dau n şiruri de caractere formate din cifre. Îmbinaţi aceste şiruri astfel, încât numărul obţinut să fie cel mai mare posibil ca valoare.
4. Se consideră două polinoame de mai multe necunoscute. Determinaţi suma acestor polinoame. De ex., P1: 3ax2-2xy+5ay+3; P2: 2ax2-3x2y+6ay+2xy. Răspuns: 5ax2+11ay+3-3x2y.
5. Între n oraşe există o reţea de drumuri ce permite ca dintr-un oraş să se ajungă în oricare alt oraş. Între două oraşe există cel puţin un drum direct de lungime cunoscută. Determinaţi traseul cu distanţa minimă posibilă ce se va parcurge la deplasarea din oraşul Ai în oraşul Aj.
6. Un şir de caractere conţine reprezentarea unui număr scris în sistemul roman de numeraţie. Determinaţi echivalentul acestui număr în sistemul zecimal de numeraţie.
7. Se consideră trei numere naturale A, B, C. Determinaţi 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.
8. Numerele a, b, c se numesc pitagoriene dacă ele pot fi lungimile laturilor unui triunghi dreptunghic. Determinaţi toate numerele pitagoriene mai mici decât o valoare dată n.
9. Fie n localităţi. Între fiecare două dintre aceste localităţi există sau nu comunicaţii aeriene . Fiind date două localităţi, să se determine dacă se poate ajunge dintr-o localitate în alta pe cale aeriană.
10. 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. Determinaţi numărul de moduri în care persoana dată poate urca scara.
11. Se dă numărul natural n. Determinaţi primele n numere naturale, a căror descompunere în factori primi conţine doar factori din mulţimea {2, 3, 5}.
12. Se consideră un dicţionar explicativ, păstrat într-u fişier text. Fiind dat un termen nou, includeţi termenul şi definiţia lui în dicţionar printr-un număr minim de operaţii.
13. Preşedintele unei ţări este ales de către parlament din care fac parte n deputaţi. Pentru a fi ales preşedintele trebuie să primească cel puţin 2/3 din voturile deputaţilor. Între anumiţi deputaţi există conflicte de interese. Doi deputaţi aflaţi în conflict de interese votează diferit. Fiind date numărul natural n şi perechile de numere x, y, unde deputaţii cu numărul de ordine x şi y au conflict de interese, să se verifice dacă este posibilă alegerea preşedintelui.
14. 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.
15. Se dau două tablouri liniare cu N (N<=500) componente numere naturale cu valori până la 1000. Determinaţi valorile ce se întâlnesc în ambele tablouri. În tablouri pot fi valori ce se repetă.
16. Scrieţi un program ce determină toate numerele naturale pseudoperfecte mai mici decât o valoare dată N. Un număr natural se numeşte pseudoperfect dacă este un divizor al sumei divizorilor săi.
17. Se consideră 9 pătrate cu laturile 2, 5, 7, 9, 16, 25, 28, 33, 36 unităţi. Asamblaţi aceste pătrate astfel încât să se obţină un dreptunghi având laturile de 61 şi 69 unităţi.
18. Se dă o matrice pătratică. Ordonaţi crescător elementele fiecărei linii, apoi rearanjaţi liniile matricei, astfel încât suma elementelor de pe diagonala principală să fie minimă.
19. Se dau M tablouri liniare cu numărul de componente diferit. Numărul de componente ale fiecărui tablou se conţine în tabloul L[1:m]. Să se determine secvenţele maxime de elemente alăturate ce figurează în toate tablourile în aceeaşi ordine.
20. 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. Determinaţi componenţa echipelor pentru cele 7 zile.
21. Se cunosc coordonatele oraşelor unei ţări. Determinaţi configuraţia unei reţele telefonice prin cablu astfel încât:a) oricare oraş să fie conectat la reţeaua telefonică; b) costul reţelei telefonice să fie minim.
22. La o întrunire trebuie să vorbească 5 persoane (A, B, C, D, E). Afişaţi toate listele de ieşire la tribună astfel încât de fiecare dată persoana B să vorbească după persoana A.
23. La o discotecă sunt M băieţi şi N fete. Determinaţi toate posibilităţile în care băieţii pot invita fetele astfel încât doi băieţi să nu invite aceeaşi fată şi fiecare băiat să danseze (N>=M).
24. Pe o hartă de contur sunt reprezentate n (n<=30) ţări. În memoria calculatorului harta este redată prin matricea [A1:n, 1:m], elementele căreia au valori 1, dacă ţările respective sunt vecine, sau 0 – în caz contrar. Determinaţi numărul minim de culori necesare pentru a colora harta astfel ca oricare două ţări vecine să aibă culori diferite. (Utilizând cel mult patru culori, coloraţi harta astfel încât două ţări cu frontieră comună să fie colorate diferit.)

|  |  |
| --- | --- |
| **Capitolul 5** | **PROBLEME CU CARACTER DE OLIMPIADĂ** |

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

**Paranteze.** Se consideră o expresie aritmetică, unde sunt folosite parantezele (,{,[,],} şi ). Elaboraţi un program care verifică dacă ‚în expresie parantezele au fost folosite corect.

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

**Output** Fişierul de ieşire *paranteze.out* va conţine pe fiecare linie cuvântul DA sau NU în funcţie de corectitudinea şirului respectiv din fişierul paranteze.txt.

**Exemplu:**

|  |  |
| --- | --- |
| Fişierul input.txt | Fişierul output.txt: |
| (a\*b)  (12-e)(d+a)[c( | DA  NU |

**Film.** *F* fete si *B* băieţi şi-au luat bilete la film. Toate biletele sunt pe acelaşi rând, pe scaune consecutive. Elaboraţi un program care să determine o posibilitate de aranjare a fetelor si băieţilor pe scaune astfel încât lângă orice băiat să se afle cel puţin o fată, iar lângă orice fată să stea cel puţin un băiat.

**Input:** Fişierul de intrare *film.in* conţine pe prima linie două numere naturale separate prin spaţiu *F B*, unde *F* reprezintă numărul de fete, iar *B* numărul de băieţi.

**Output:** Fişierul de ieşire *film.out* va conţine o singura linie pe care vor fi scrise *F* litere *'f'* si *B* litere *'b*' (*'f*' indicând o fată, iar *'b'* indicând un băiat), respectând condiţiile problemei.

**Restricţii:** 1 <= *F, B* <= 100

Pentru datele de test exista întotdeauna soluție.

**Exemplu:**

|  |  |
| --- | --- |
| *film.in* | *film.out* |
| 3 4 | bfbfbfb |

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

**Input:** Fişierul de intrare *finala.in* conţine pe prima linie trei numere naturale separate prin câte un spaţiu *N M K*, cu semnificaţia din enunţ.

**Output:** Fişierul de ieşire *finala.out* va conţine un singur număr natural reprezentând numărul minim de autobuze necesare. Dacă nu este posibilă organizarea transportului în condiţiile problemei, veţi afişa valoarea 0.

**Restricţii:** 1<=*N, M, K*<=10000

**Exemplu:**

|  |  |  |
| --- | --- | --- |
| *finala.in* | *finala.out* | ***Explicaţii*** |
| 10 4 7 | 2 | Sunt 10 copii şi 4 profesori. În fiecare autobuz vor merge 5 copii şi 2 profesori, deci sunt necesare 2 autobuze |
| 10 4 5 | 0 | Transportul nu poate fi organizat în condiţiile date |

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

Există *n* ţări care exportă petrol. Fiecare *i*-stat exportă petrol cu *ai* dolari şi *bi*euro pentru un baril. Moldova nu poate procura petrol de la un stat şi în dolari şi în euro. Ajutaţi Republica Moldova să procure petrol cât mai mult, dacă se cunoaşte numărul *a*(numărul de dolari de care dispune RM) şi *b*(numărul de euro de care dispune RM).

**Input:** Fişierul de intrare *oil.in* conţine în primul rând 3 numere: *n, a, b* (1 ≤ n ≤ 100, 0 ≤ a, b ≤ 1000). Următoarele *N* rânduri conţin perechile de numere *ai* şi *b*i

Output: Fișierul de ieşire oil.out va conţine 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 şi operatorii binari +, –, \**.* Scrieţi un program care evaluează expresiile aritmetice în studiu.

## Input: Fişierul input.txt va conţine expresia de evaluat.

**Output**: Fişierul *output.txt* va conţine rezultatul evaluării.

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

**Exemplu**:

|  |  |
| --- | --- |
| Fişierul input.txt | Fişierul output.txt: |
| 16+34+2\*4–7 | 51 |

**Parcurgerea matricei.** Se dă matricea **A[N,N]** de numere întregi (N≤10). De parcurs matricea în următoarea ordine: primul se va afişa elementul **A[N,1],** urmează mişcarea în dreapta, apoi paralel cu diagonala principală, mişcarea în sus, apoi iarăşi paralel cu diagonala principală, ş.a.m.d., ultimul element va fi **A[1,N]**.

**Input:** De la tastatură se introduce N şi elementele matricei

**Output:** La ecran se vor afişa 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 conţine doar litere mici ale alfabetului englez. Textul poate fi codificat înlocuind apariţiile consecutive ale subşirurilor sale de lungime maximă cu subşirul urmat de numărul său de apariţii. Scrieţi un program care să codifice optimal un text dat, adică codul rezultat să fie de lungime minimă.

**Input:** fişierul *cod.in* conţine pe o linie textul.

**Output:** fişierul *cod.out* va conţine codul.

**Exemplu:**

|  |  |
| --- | --- |
| Fişierul cod.in | Fișierul cod.out |
| aaacaaacaaacbbdefdef | aaac3b2def2 |

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

11235813213455...

**Cerinţă**: Se cere să scrieţi un program care să determine cifra de pe poziţia N din şirul obţinut (poziţiile în şir sunt numerotate începând cu 1).

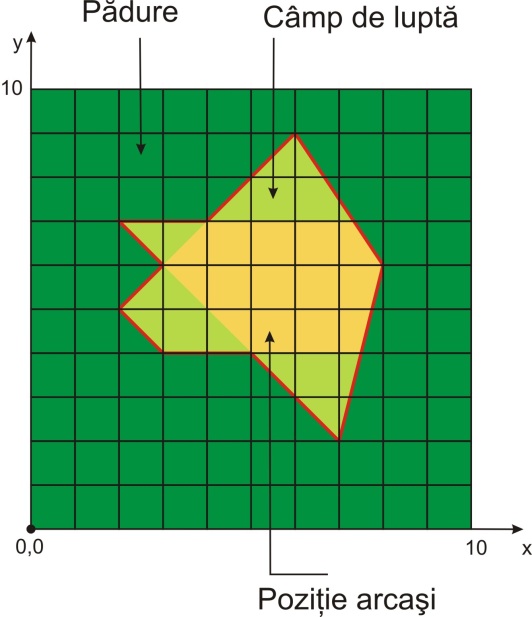
Date de ieşire: va fi afişat un număr natural – valoarea cifrei de pe poziţia N.

Restricţii

1 ≤ N ≤ 10000000

**Exemple**:

|  |  |  |
| --- | --- | --- |
| **Date de intrare** | **Date de ieşire** | **Explicaţii** |
| 6 | 8 | 112358 |
| **Date de intrare** | **Date de ieşire** | **Explicaţii** |
| 14 | 5 | 11235813213455 |



**Arcaşi**[12] Secretul victoriilor faimosului comandant de oşti MegaFlop este strategia lui de alegere a poziţiei arcaşilor pe câmpul de luptă. Câmpul de luptă are forma unui poligon simplu şi e înconjurat de păduri. MegaFlop plasează arcaşii doar pe poziţii din care este **văzut** tot câmpul de luptă. Se consideră că arcaşii văd tot câmpul, dacă din orice punct care aparţine poziţiei lor de tragere se poate trage cu săgeata în orice alt punct al câmpului. Traiectoria săgeţii este liniară. Nimerind în pădure, săgeata se pierde. Pentru tragere, fiecare arcaş are nevoie de 1 unitate de suprafaţă. Astfel, numărul maxim de arcaşi, care pot fi plasaţi pe poziţii este determinat de aria poligonului din care este văzută toată câmpia. (des. 1)

**Cerinţă.** Scrieţi un program, care determină numărul maxim de arcaşi, care pot fi plasaţi pe poziţii pe câmpul de luptă.

**Input.** Fişierul de intrare va conţine pe prima linie un număr întreg N - numărul de vârfuri ale poligonului simplu, care descrie perimetrul câmpului de luptă. Urmează N linii care conţin coordonatele vârfurilor poligonului în ordinea parcurgerii lor după acele de ceasornic, câte un vârf pe linie. Linia *i+*1 conţine două numere întregi *xi*, *yi*, separate prin spaţiu – coordonatele vârfului *i*.

**Output.** Fişierul de ieşire va conţine un singur număr întreg: numărul maxim de arcaşi, care pot fi plasaţi pe poziţii.

**Restricţii**

3 ≤ N ≤ 1 000, 0< *xi*, *yi*, ≤ 10000

**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 |

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

**Cerinţă.** Scrieţi un program, care va determina din câte cetăţi vor demobilizate garnizoanele.

**Date de intrare.** Prima linie a fişierului de intrare cetati.in conţine un număr întreg: *N* (3 *≤ N ≤* 10000) – numărul de cetăţi în Terralanda.

Urmează *N* linii ce conţin câte două numere întregi *xi,yi*   
(–109 *≤ xi, yi ≤* 109), separate prin spaţiu – coordonatele cetăţilor.

**Date de ieşire.** Fişierul de ieşire cetati.out va conţine un număr întreg – numărul de cetăţi 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 | img |

**Pătrate**[12] Fie dată o reţea din puncte cu coordonate întregi *N* × *N*. Fie unele noduri ale reţelei sunt colorate cu alb, altele – cu negru. Se cere să determinaţi numărul de pătrate pe reţeaua dată (vârfurile unui pătrat trebuie să coincidă cu nodurile reţelei şi să fie colorate cu aceeaşi culoare).

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

|  |  |
| --- | --- |
|  |  |
| Des 1. Reţea 4 × 4. | Des 2. Pătratul din reţea. |

Input. Prima linie a fişierului de intrare conţine numărul N – dimensiunea reţelei (2 ≤ N ≤ 50). Următoarele N linii conţin câte N numere din mulţimea {0, 1} şi descriu reţeaua. 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. Fişierul de ieşire va conţine un singur număr – cel al pătratelor din reţea.

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 şir S de simboluri (simbolurile sunt numerotate de la stânga la dreapta, începând cu indicele 1) este lovit în poziţia **i** de o ghiulea. Imediat după lovitură se produce „explozia”, după următoarea regulă:

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

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

* Dacă simbolurile din poziţiile **i-1, i, i+1** sunt egale, din şir se lichidează toată secvenţa de simboluri egale, care le conţine, iar fragmentele rămase se unesc.

*Exemplu:* **BBAAAAACCC** este lovit în poziţia 4 *Rezultat:* **BBCCC**

* Atât timp cât după explozie în locul unirii fragmentelor se formează o secvenţă de cel puţin 3 litere egale, explozia se repetă.

*Exemplu*: **DBBAAAAABCCC** este lovit în poziţia 6

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

***Cerinţă*:** Scrieţi un program, care să determine şirul în care se transformă şirul iniţial S după lovitura unei ghiulele.

***Restricţii***: Lungimea S nu va depăşi 200 caractere. 1 < i < lungimea şirului S

***Input*:** Fişierul de intrare x-**zuma.in** va conţine pe prima linie un număr natural – valoarea **i** a poziţiei în care loveşte ghiuleaua. Linia a doua a fişierului conţine şirul S

***Output*:** Fişierul de ieşire x-**zuma.out** va conţine pe prima (şi unica) linie şirul S după lovitură şi explozii.

|  |  |
| --- | --- |
| **zum.in** | **zum.out** |
| 12  AABBCCCCBRQQQAA | AABBCCCCBRAA |
| 6  AABBCCCCBRQQQAA | AARQQQAA |
| 5  AABBCCCCBRQQQAA | AABBCCCCBRQQQAA |

***Exemplu*:**

**Mere**. În grădina lui Ion creşte 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 şi începe să cadă vertical în jos. Dacă în timpul căderii mărul atinge un alt măr, acesta din urmă începe să cadă şi el vertical în jos. Traiectoria merelor în cădere nu se schimbă la atingerea altor mere. Astfel, orice măr, cu excepţia primului, începe să cadă doar după ce a fost atins de un măr în cădere.

**Cerinţă** Se cere să scrieţi un program, care să determine numărul de mere, care vor cădea din pom.

**Date de intrare.** Fişierul de intrare mere.in va conţine pe prima linie numărul n al merelor din pom. Următoarele n linii conţin descrierea merelor. Linia i+1 conţine 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, codiţa fiind punctiformă) xi , y i şi zi şi raza r i toate numerele - întregi). Se garantează, că iniţial nici care două mere nu se intersectează. Axa OZ este orientată vertical în sus.

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

**Restricţii**

 1 ≤ N ≤ 200

 -10000 ≤ (xi, yi, zi) ≤ 10000

 1 ≤ ri ≤ 10000

Exemplu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **mere.in** | **mere.out** | **Explicaţii** | | 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 şi mărul cu indicele 4. | |

|  |  |
| --- | --- |
| **Capitolul 6** | **INFORMAȚII UTILE ÎN TIMPUL LUCRULUI ÎN MEDIUL TURBO PASCAL** |

**CODURILE GENERATE DE TASTATURĂ**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tasta** | **Normal** | **Shift** | **Ctrl** |  | **Tasta** | **Normal** | **Shift** | **Ctrl** |
| A | 97 | 65 | 1 |  | `~ | 96 | 126 |  |
| B | 98 | 66 | 2 |  | -\_ | 45 | 95 |  |
| C | 99 | 67 | 3 |  | =+ | 61 | 43 |  |
| D | 100 | 68 | 4 |  | [{ | 91 | 123 |  |
| E | 101 | 69 | 5 |  | ]} | 93 | 125 |  |
| F | 102 | 70 | 6 |  | \| | 92 | 124 |  |
| G | 103 | 71 | 7 |  | ;: | 59 | 58 |  |
| H | 104 | 72 | 8 |  | ' " | 39 | 34 |  |
| I | 105 | 73 | 9 |  | ,< | 44 | 60 |  |
| J | 106 | 74 | 10 |  | .> | 46 | 62 |  |
| K | 107 | 75 | 11 |  | /? | 47 | 63 |  |
| L | 108 | 76 | 12 |  | Backspace | 8 | 8 | 127 |
| M | 109 | 77 | 13 |  | Tab | 9 | 0,15 |  |
| N | 110 | 78 | 14 |  | Enter | 13 | 13 | 10 |
| O | 111 | 79 | 15 |  | Spațiu | 32 | 32 | 32 |
| P | 112 | 80 | 16 |  | Esc | 27 | 27 | 27 |
| Q | 113 | 81 | 17 |  | F1 | 0,59 | 0,84 | 0,94 |
| R | 114 | 82 | 18 |  | F2 | 0,60 | 0,85 | 0,95 |
| S | 115 | 83 | 19 |  | F3 | 0,61 | 0,86 | 0,96 |
| T | 116 | 84 | 20 |  | F4 | 0,62 | 0,87 | 0,97 |
| U | 117 | 85 | 21 |  | F5 | 0,63 | 0,88 | 0,98 |
| V | 118 | 86 | 22 |  | F6 | 0,64 | 0,89 | 0,99 |
| W | 119 | 87 | 23 |  | F7 | 0,65 | 0,90 | 0,100 |
| X | 120 | 88 | 24 |  | F8 | 0,66 | 0,91 | 0,101 |
| Y | 121 | 89 | 25 |  | F9 | 0,67 | 0,92 | 0,102 |
| Z | 122 | 90 | 26 |  | F10 | 0,68 | 0,93 | 0,103 |
| 0) | 48 | 41 |  |  | PrintScr |  |  | 0,114 |
| 1! | 49 | 33 |  |  | Insert | 0,82 | 0,5 | 0,4 |
| [2@](mailto:2@) | 50 | 64 |  |  | Home | 0,71 | 0,71 | 0,119 |
| 3# | 51 | 35 |  |  | PageUp | 0,73 | 0,73 | 0,132 |
| 4$ | 52 | 36 |  |  | Delete | 0,83 | 0,7 | 0,6 |
| 5% | 53 | 37 |  |  | End | 0,79 | 0,79 | 0,117 |
| 6^ | 54 | 94 |  |  | PageDn | 0,81 | 0,81 | 0,118 |
| 7& | 55 | 38 |  |  |  |  |  |  |
| 8\* | 56 | 42 |  |  |  |  |  |  |
| 9( | 57 | 40 |  |  |  |  |  |  |

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 inexistenta**. 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 inexistenta**. 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 procedu-rile *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 neasignat**. 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/funcții, 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 procedu-rile *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 Ivaşc, Mona Prună, Tehnici de programare (aplicaţii de laborator) Ed. Petrion, Bucureşti, 2000
3. Culegere de problem de informatică, Gh. Bostan, Editura Lumina, Chișinău, 1996;
4. FIorin Munteanu, Traian lonescu, Daniela Tătaru, Sergiu Mihai Dascala, Gheorghe Muscă., Programarea calculatoarelor, manual pentru licee de informatică clasele X-XII, Ed. Didactică şi Pedagogică, R.A, - Bucureşti, 1994
5. Ilie Coandă, Pascal programe-exemple LITERA, Chişină -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, Bucureşti, Corint, 2000
8. Ion Smeureanu, Ion Ivan, Mmarian Dârdală., Limbajul C/C++ prin exemple, Ed. CISON, Bucureşti, 1996
9. Tudor Sorin, Informatica(Tehnici de programare), Varianta C++, manual pentru clasa a X, Ed. L&S, Bucureşti
10. Tudor Sorin, Informatica, Varianta C++, manual pentru clasa a IX, Ed. L&S, Bucureşti
11. Tudor Sorin, Informatica, Varianta C++, manual pentru clasa a XI, Ed. L&S, Bucureşti
12. http://campion.edu.ro/