

Colegiul Dobrogean „SPIRU HARET” Tulcea

INSTRUCȚIUNEA ALTERNATIVĂ

Autor:

Prof. MUSCALU ADRIAN



CUPRINS

Nr. crt.	Conținut	Pagina
1.	Instrucțiunea alternativă- generalități	3
2.	Instrucțiunea if else	3
3.	Schema logică	11
4.	Probleme: Instrucțiunea alternativă	12
5.	Bibliografie	17

INSTRUCȚIUNEA ALTERNATIVĂ

Instrucțiunile de control

Instrucțiunile dintr-un program se execută în ordinea în care le-ați scris (secvențial). În marea majoritate a algoritmilor nu se execută toți pașii secvențial. Sunt frecvente cazurile în care se execută anumiți pași în funcție de o anumită condiție sau se repetă executarea anumitor pași atât timp cât este îndeplinită o anumită condiție. Pentru a putea implementa acest gen de algoritmi, în limbajul de programare se folosesc instrucțiunile de control. La fel ca și în cazul unui algoritm exprimat în pseudocod, ele specifică ordinea în care se execută instrucțiunile programului, controlând fluxul de execuție al programului.

Instrucțiunile de control definite în limbajul C++ sunt:

1. Structura alternativă:

- ✓ simplă - if else
- ✓ generalizată - switch case.

2. Structura repetitivă:

- ✓ condiționată anterior - while, for
- ✓ condiționată posterior - do while

INSTRUCȚIUNEA IF ELSE

Instrucțiunea if else implementează structura alternativă simplă (dacă... , atunci... altfel... sfârșit_dacă). Sintaxa acestei instrucțiuni este:

Pseudocod

```
dacă condiție
atunci
acțiune 1;
[altfel acțiune_2; ]
sfârșit__dacă;
```

Limbajul C++

```
if (expresie)
    instrucțiune_1
[else instrucțiune_2 ]
```

Expresia trebuie să furnizeze un rezultat numeric care va fi interpretat ca o constantă logică: True sau False. Instrucțiunea if else se execută astfel:

Pas 1. Se evaluează expresia expresie.

Pas 2. Dacă rezultatul expresiei este diferit de 0 (corespunde valorii logice True), se execută instrucțiune_1. Dacă rezultatul expresiei este 0 (corespunde valorii logice False), se execută instrucțiune_2.

Exemplul 1:

Programul verifică dacă un număr întreg citit de la tastatură este un număr par:

```
#include <iostream.h>
void main()
{int x;
cout<<"x="; cin>>x;
  if (n%2==0)
cout<<"Numărul este par ";
else
cout<<"Numărul nu este par ";
}
```

Pentru structura alternativă simplă cu o ramură vidă nu se mai precizează else instrucțiune 2, instrucțiunea if având în acest caz sintaxa:

```
if (expresie) instrucțiune
```

Exemplul 2:

Programul afișează modulul unui număr real citit de la tastatură

```
#include <iostream.h>
void main()
{int x;
  cout<<"x="; cin>>x;
if (x<0) // x<0 este expresia
x=-x; // instrucțiune
  cout<<"modulul="<<x; )
```

Observații:

1. Conform sintaxei instrucțiunii if, pe fiecare dintre ramurile structurii alternative nu se poate executa decât o singură instrucțiune. În cazul în care în algoritmul pentru rezolvarea problemei este nevoie să se execute mai mulți pași pe oricare dintre ramuri, înseamnă că și în instrucțiunea if trebuie să se execute mai multe instrucțiuni pe acea ramură, și nu una singură. Aceste cazuri se rezolvă prin încapsularea grupului de instrucțiuni într-o instrucțiune compusă.

Exemplul 3

Programul afișează în ordine crescătoare două numere reale a și b citite de la tastatură, folosind algoritmul de interschimbare a două variabile de memorie.

```

#include <iostream.h>
void man()
{
float a,b,x;
cout<<"a="; cin>>a;
cout<<"b="; cin>>b;
if (b<a)
{x=a; a=b; b=x;} // instrucțiunea compusă
cout<<a<<" " <<b;
}

```

Deoarece în limbajul C++ nu este implementat tipul logic, orice valoare numerică poate fi interpretată ca o constantă logică (True sau False). În cazurile în care în expresie se testează dacă o valoare este diferită de 0 (orice număr diferit de zero înseamnă True), expresia poate fi scrisă condensat.

Exemplul 4:

Programul verifică dacă un număr întreg citit de la tastatură este impar.

```

#include <iostream.h>
void main()
{int n;
cout<<"n="; cin>>n;
if(n%2)
cout<<"Numarul este impar";
else
cout<<"Numarul nu este im par";
}

```

În același mod se poate scrie și expresia pentru testarea unui număr par. Dacă $n\%2$ trebuie să fie egal cu 0 pentru ca numărul să fie par, înseamnă că $!(n\%2)$ trebuie să fie egal cu 1, adică diferit de 0 și, în loc să scriem expresia $!(n\%2)\neq 0$, putem condensa expresia cu $!(n\%2)$.

3. Pentru a implementa unii algoritmi se pot folosi instrucțiuni if imbricate. Să considerăm următorul exemplu.

Exemplul 5:

Se introduc de la tastatură numerele întregi a și b și un caracter care reprezintă o operație aritmetică. Să se calculeze valoarea lui e definită ca rezultat al aplicării operatorului aritmetic pe numerele a și b.

```

#include <iostream. h>
void main()
{ int a,b;
char c;
float e;
}

```

```

cout<<"a="; cin>>a;
cout<<"b="; cin>>b;
  cout<<"c="; cin>>c;
if (c=='+')
{e=a+b; cout<<"E="<<e;}
  else
if(c=='-')
{ e=a-b;   cout<<"E="<<e;}
  else
if(c=='*')
{ e=a-b;   cout<<"E="<<e;}
  Else
if (c=='/')
{e=(float) a/b; cout<<"E= " <<e; }
  else cout<<"operator gresit";
}

```

Construcțiile în care folosiți instrucțiuni if imbricate pot crea însă probleme atunci când una dintre structurile alternative este cu o ramură vidă.
Să considerăm următorul exemplu.

Exemplu 6:

Se citesc trei numere întregi, n , a și b . Să se calculeze valoarea lui x , astfel: dacă n este diferit de 0, x are valoarea lui, altfel, are valoarea lui b .

Să presupunem că se scrie următorul program:

```

include <iostream .h>
void main()
{
int a,b,n,x;
cout<<"n="; cin>>n;
cout<<"a="; cin>>a;
cout<<"b="; cin>>b;
  x=a;
  if (n)
  if (a<0) x=-a
else x=b;
  cout<<"x= " <<x;
}

```

Scrierea indentată a acestor instrucțiuni nu rezolvă problema de sintaxă a limbajului. Conform sintaxei limbajului, cuvântul cheie else cel mai interior este asociat întotdeauna cu cuvântul cheie if cel mai interior. Așa cum a fost scris acest program, instrucțiunea $x=b$; se va executa dacă a este mai mare sau egal cu 0, și nu dacă n este diferit de 0. Pentru a corecta această greșeală, vom încapsula, într-o instrucțiune compusă,

instrucțiunea care se execută atunci când n este 0, chiar dacă aparent nu este necesar, fiind vorba de o singură instrucțiune.

Exemplul 7

```
#include < void main()
{
int a,b,n ;
cout<<"n="; cin>>n;
cout<<"a="; cin>>a;
cout<<"b="; cin>>b;
if (n)
{if (a<0) x=-a;} // instrucțiunea compusă
else
cout<<"x= " <<x;
}
```

Instrucțiunea switch case

Instrucțiunea switch case implementează structura alternativă generalizată (în caz că... cazul... altfel... sfârșit?in_caz_că). Sintaxa acestei instrucțiuni este:

Pseudocod

```
În caz că selector
cazul v 1: acțiune 1;
cazul v2: acțiune 2;
.....
cazul vi: acțiune i;
.....
cazul vn: acțiune n;
altfel acțiune n+1;
sfârșit in_caz_că;
```

Limbaajul C++

```
switch (expresie)
case exp_1: instrucțiune1 break;
case exp2: instrucțiune 2 break;
.....
case exp_i: instrucțiune_i break;
.....
case exp_n: instrucțiune_n break;
[default: instrucțiune_n+1]
```

Expresia expresie trebuie să furnizeze un rezultat numenc întreg, iar expresia care se evaluează pentru fiecare caz (exp_i) trebuie să fie constante întregi sau expresii constante cu valoare întregă. Caracterul : este un separator între eticheta cazului (case exp_i) și instrucțiunea care se va executa (instrucțiune i). Colecția de instrucțiuni etichetate este încapsulată într-un bloc delimitat de separatorii Instrucțiunea switch ... case se execută astfel:

Pas 1. Se evaluează expresia expresie

Pas 2. Dacă rezultatul obținut are valoarea exp_1 se execută instrucțiune_1 după care se trece la execuția instrucțiunii care urmează instrucțiunii switch, altfel se trece la pasul următor.

Pas 3. Dacă rezultatul obținut are valoarea exp_2 se execută instrucțiune_2 după care se trece la execuția instrucțiunii care urmează instrucțiunii switch, altfel se trece la pasul următor.

Pas n+1. Dacă rezultatul obținut are valoarea exp_n se execută instrucțiune_n după care se trece la execuția instrucțiunii care urmează instrucțiunii switch, altfel se execută instrucțiunea care urmează după eticheta default, dacă aceasta există, după care se trece la execuția instrucțiunii care urmează instrucțiunii switch.

Exemplul 8:

Se introduc de la tastatură numerele întregi a și b și un caracter care reprezintă o operație aritmetică. Să se calculeze valoarea lui e definită ca rezultat al aplicării operatorului aritmetic pe numerele a și b.

```
#include <iostream.h>
void main()
{int a, b;
char c;
float e;
  cout<<"a="; cin>>a;
  cout<<"b="; cin>>b;
  cout<<"c="; cin>>c; s
  switch (c)
  { // începe blocul instrucțiunii switch.
    case '+': {e=a+b; cout<<"E= "<<e;} break;
    case '-': {e=a-b;-cout<<"E= "<<e;} break;
    case '*': {e=a*b; cout<<"E=' "<<e;} break;.
    case '/': {e=(float)a/b; cout<<"E= "<<e; } break;
    default: cout<<"Operator greșit";
  } //închide blocul instrucțiunii switch
```

Observații:

1. Toate expresiile corespunzătoare cazurilor (exp_i) trebuie să fie diferite între ele.
2. Eticheta default este opțională. Instrucțiunea atașată acestei etichete se execută numai dacă nu a fost îndeplinit nici un caz anterior.
3. Instrucțiunea break; este obligatorie. Semnificația ei este de a se întrerupe execuția instrucțiunii switch case și de a se trece la executarea următoarei instrucțiuni din program. Dacă nu scrieți această instrucțiune, se continuă execuția instrucțiunii switch ... case. Deoarece cuvintele cheie case nu sunt decât niște etichete folosite pentru evidențierea cazurilor, continuarea execuției instrucțiunii switch case înseamnă de fapt

executarea tuturor instrucțiunilor până la întâlnirea separatorului care închide blocul instrucțiunii switch ... case.

Exemplul 9:

```
<include <iostream.h>
void main()
{int a,b;
  int n;
  cout<<"n =(1,2 sau 3)"; cin>>n;
  switch (n)
  {
  case 3: cout<<3<<" ";
  case 2: cout<<2<<" ";
  case 1: cout<<1<<" "; break;
  default: cout<<"Ati ales alt nurnar";
  }
}
```

La execuția acestui program, în funcție de valoarea citită pentru n, se afișează:

✓ Dacă n are valoarea 3, se execută instrucțiunea cu eticheta case 3, după care se continuă cu execuția instrucțiunii cu eticheta case 2, și apoi cu a instrucțiunii cu eticheta case 1, după care se execută instrucțiunea break care întrerupe execuția instrucțiunii switch ... case. Programul va afișa 3 2 1.

✓ Dacă n are valoarea 2, se execută instrucțiunea cu eticheta case 2, după care se continuă cu execuția instrucțiunii cu eticheta case 1, și apoi cu execuția instrucțiunii break, care întrerupe execuția instrucțiunii switch ... case. Programul va afișa 2 1.

✓ Dacă n are valoarea 1, se execută instrucțiunea cu eticheta case 1, apoi instrucțiunea break, care întrerupe execuția instrucțiunii switch case. Programul va afișa 1.

✓ Dacă n are o valoare diferită de 1, 2 sau 3, se execută instrucțiunea cu eticheta default. Programul va afișa Ați ales alt număr.

Observație. Instrucțiunea switch ...case poate fi folosită în locul unor instrucțiuni if else imbricate. Să considerăm următorul exemplu:

Exemplul 10:

Programul citește două numere întregi a și b și un caracter c. Dacă pentru c se citește 1 sau 3, se afișează suma celor două numere. Dacă pentru c se citește 2 sau 4, se afișează diferența dintre valoarea mai mare și valoarea mai mică a celor două numere. În restul cazurilor se afișează mesajul. - Altă opțiune.

```
#include <iostream.h>
void main()
{int a,b;
  char c;
  cout<<"a= ; cin>>a;
```

```

    cout<<"b= cin>>b;
cout<<"optiune (1, 2,3 sau 4 )= "; cin>>c;
if(c==1 || c==3)
cout<<a+b;
else
if(c==2 || c==4)
if (a>b) cout<<a-b;
else
cout<<b-a;
else
cout<<"Alta optiune";
}

```

Programul poate fi implementat și cu o structură switch case, astfel:

Exemplul 11:

```

#include <iostream.h>
void main( )
{ int a,b ;
char c;
cout<<"a=" ; cin>>a;
cout<<"b= "; cin>>a;
cout<<"optiune (1 , 2, 3 sau 4)= "; cin>>c;

switch (c )
{
case '1': case '3': cout<<a+b; break;
case '1': case '3':
if (a>b) cout<<a-b;
else
cout<<b-a;
default: cout<<"Altă opțiune";
}
}

```

În funcție de valoarea citită pentru *c*, instrucțiunea switch case se execută astfel:

✓ Dacă *c* are valoarea '1', se execută instrucțiunea cu eticheta case '1', care este o instrucțiune inexistentă (nu se execută nimic), după care se continuă cu execuția instrucțiunii cu eticheta case '3', prin care se afișează suma dintre *a* și *b*, și apoi se execută instrucțiunea break, care întrerupe execuția instrucțiunii switch case. Dacă *c* are valoarea '3', se execută direct instrucțiunea cu eticheta case '3', obținându-se același rezultat ca și în cazul valorii '1' pentru *c*.

✓ Dacă *c* are valoarea '2', se execută instrucțiunea cu eticheta case '2', care este o instrucțiune inexistentă (nu se execută), după care se continuă cu execuția instrucțiunii cu eticheta case '4', prin care se afișează diferența dintre valoarea mai mare și valoarea mai mică a numerelor *a* și *b* (instrucțiunea if else), și apoi se execută instrucțiunea break, care întrerupe execuția instrucțiunii switch ... case. Dacă *c* are valoarea '4', se execută

direct instructiunea cu eticheta case '4', obtinandu-se același rezultat ca și în cazul valorii '2' pentru c.

✓ Dacă c are o valoare diferită de '1', '2', '3' sau '4', se execută instructiunea cu eticheta default și se va afișa Altă opțiune.

SCHEMA LOGICĂ

În schema logică, instructiunea alternativă este reprezentată printr-un romb.

Exemplul 12:

Să se scrie algoritmul în C++ și schema logică pentru aflarea maximului dintre trei numere întregi a,b,c.

Pseudocod	Schema logică
<pre> include <iostream> int main() { int a,b,c; cin>>a>>b>>c; if(a>b) if(a>c) cout<<a; else cout<<c; else if(b>c) cout<<b; else cout<<c; return 0; } </pre>	<pre> graph TD START([START]) --> READ[/CITEȘTE a,b,c/] READ --> D1{a > b} D1 -- DA --> D2{a > c} D1 -- NU --> D3{b > c} D2 -- DA --> PRINT_A[/SCRIE a/] D2 -- NU --> PRINT_C1[/SCRIE c/] D3 -- DA --> PRINT_B[/SCRIE b/] D3 -- NU --> PRINT_C2[/SCRIE c/] PRINT_A --> J1(()) PRINT_C1 --> J1 PRINT_B --> J2(()) PRINT_C2 --> J2 J1 --> J2 J2 --> STOP([STOP]) </pre>

PROBLEME: INSTRUCȚIUNEA ALTERNATIVĂ

- 1) Se citește un număr natural. Să se afișeze mesajul "Este cifra" dacă numărul este mai mic decât 10 și "Nu e cifră" , în caz contrar.
- 2) Se citește un număr natural. Să se afișeze un mesaj după cum numărul este par sau impar.
- 3) Se citește un caracter. Să se afișeze mesajul "este literă mică" sau "este literă mare" după caz. Să se modifice programul astfel încât să afișeze mesajul "nu este literă" dacă acest lucru este adevărat.
- 4) Se citește un număr care reprezintă luna din an (1 pentru ianuarie, 2 pentru februarie etc.). Afișați numărul lunii precedente și numărul lunii următoare.
- 5) Un număr se numește palindrom dacă citit de la coadă la cap este același. De exemplu 121 este palindrom pe când 1234 nu este. Se citește un număr de 3 cifre. Să se afișeze un mesaj dacă este palindrom sau nu.
- 6) Care este cea mai mare valoare pe care o poate avea variabila întreaga k pentru ca instrucțiunea `if (k<8-3*k) cout<< "Mesaj";` să afișeze mesajul?
- 7) Se citesc trei numere naturale. Să se determine câte numere sunt pare.
- 8) Să se scrie un program care determină maximul a două numere întregi citite de la tastatură.
- 9) Să se scrie un program care citește de la tastatură trei numere naturale și determină diferența dintre cel mai mare și cel mai mic.
- 10) Se citește de la tastatură un număr natural de 3 cifre. Să se determine câte cifre impare conține.
- 11) Să se scrie un program care verifică dacă un an calendaristic precizat este bisect. Anii bisecți sunt cei divizibili cu 4, dar nedivizibili cu 100 sau divizibili cu 400.
- 12) În fiecare zi lucrătoare din săptămână, Pinocchio spune câte o minciună datorită căreia nasul acestuia crește cu câte p cm pe zi. Sâmbăta și duminica, când vine bunicul Geppetto acasă, pentru a nu-l supăra prea tare, Pinocchio reușește să nu spună nici o minciună, ba chiar, uitându-se în oglindă, observă că în fiecare din aceste zile lungimea nasului său scade cu câte 1 cm pe zi. Când începe o nouă săptămână, rămânând singur acasă, Pinocchio continuă șirul minciunilor. Care este dimensiunea nasului lui Pinocchio după k zile (zilele încep cu luni), știind că inițial nasul său măsura n cm? Exemplu: pentru $n=2$ $p=1$, $k=8$, se va afișa 6 cm.

- 13) Se dau două numere naturale nenule n și k . Să se determine multiplul lui k cel mai apropiat de n .
- 14) Se citesc de la tastatură 3 numere naturale, a, b, c . Să se realizeze cu a și b operația codificată prin numărul c .
- 15) Se citește de la tastatură un număr natural de 3 cifre. Să se afișeze pe ecran cea mai mare cifră a sa.
- 16) Se citesc de la tastatură trei numere naturale. Să se determine numărul de valori distincte.
- 17) Se citește un număr natural n . Să se stabilească dacă n poate fi scris ca sumă de trei numere naturale consecutive.
- 18) Se citesc 3 numere naturale a, b, c . Să se afișeze cele 3 numere în ordine crescătoare a valorii lor.

19) Fie următoarea secvență:

```

if (a>=b)
    if(a>7  && b<4)
        x = a*b;
    else x = a+b;
else x = b-a;
cout<<x;

```

Ce valoare afișează secvența dacă a) $a = 8, b = 5$ b) $a=2, b=3$?

20) Se introduc vârstele a doi copii. Afișați care copil este mai mare și diferența de vârstă dintre cei doi. Exemplu :

Date de intrare : 6 13 date de ieșire : al doilea copil este mai mare cu 7 ani.

21) Dintr-o cutie cu trei numere se extrag două numere. Cunoscând suma celor două numere extrase, să se afișeze numărul rămas în cutie. Exemplu : date de intrare : numere existente în cutie 5 12 8, suma numerelor extrase 13, date de ieșire : 12.

22) Se introduc două numere întregi. Să se testeze dacă primul număr este predecesorul (succesorul) celui de-al doilea și să se afișeze un mesaj corespunzător. Exemple : date de intrare : 2 4 , date de ieșire : Nu ; date de intrare : 5 6 , date de ieșire : Da.

23) Un lift pentru copii acceptă o greutate de maxim 100 kg. Introducând în calculator greutățile a doi copii, specificați prin afișarea unui mesaj corespunzător, dacă POT INTRA AMBII COPII sau dacă INTRĂ PE RÂND.

Exemple : Date de intrare : greutăți copii: 87 50 Date de ieșire : Intră pe rând

Date de intrare : 45 52 ; Date de ieșire: Pot intra ambii copii.

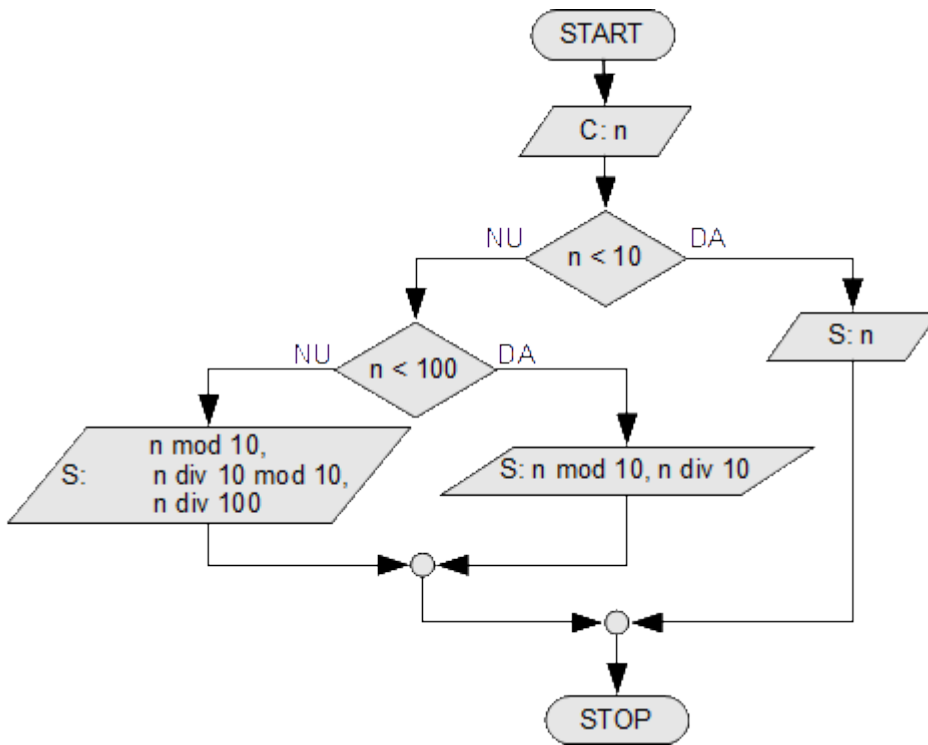
24) Se citesc două numere naturale n, m cu exact trei cifre fiecare. Să se afle câte cifre ale lui n divid pe m .

25) Se introduc trei numere. Să se verifice dacă formează o secvență de numere consecutive. Exemple:

Date de intrare: 3 4 5 Date de ieșire: Da

Date de intrare; 4 5 7 Date de ieșire: Nu.

26) Ce afișează următoarea schemă logică, dacă se citesc numerele 7, 47 sau 536?



27) Ionel spune părinților doar notele mai mari sau egale cu 7. Într-o zi, el a luat trei note. Introduceți-le în calculator și afișați acele note pe care le va comunica și părinților. Exemplu : Date de intrare: 8 7 5 Date de ieșire; 8 7.

28) Pe o masă de biliard sunt bile albe, roșii și verzi. Din fiecare culoare sunt bile de două dimensiuni: mari și mici. Să se afișeze câte bile sunt în total pe masa de biliard. Un jucător vrea să-i spuneti care bile sunt mai multe, cele mici sau cele mari, afișând numărul lor. Ce culoare au bilele cele mai numeroase? Precizați numărul lor.

Exemplu: Date de intrare: Nr. bile albe mici: 2 ; Nr. bile albe mari: 3 ; Nr. bile roșii mici: 1
Nr. bile roșii mari:4 ; Nr. bile verzi mici: 3 ; Nr. bile verzi mari: 4

Date de ieșire: Totalul bilelor: 17 Mari : 11 bile Verzi: 7 bile

29) Într-o tabără, băieții sunt cazați câte 4 într-o căsuță, în ordinea sosirii. Ionel a sosit al n-lea. În a câta căsuță se va afla? Exemplu : date de intrare : n=69 date de ieșire : căsuța 17.

30) Cunoscând data curentă exprimată prin trei numere întregi reprezentând anul, luna, ziua precum și data nașterii unei persoane, exprimată la fel, să se facă un program care să calculeze vârsta persoanei respective exprimată în număr de ani împliniți. Exemplu :

Date de intrare : data curentă 2005 10 25, data nașterii 1960 11 2, Date de ieșire 44 ani.

31) Se dă următorul algoritm scris în pseudocod. Să se precizeze ce execută acest algoritm pentru $n=5$.

```
intreg p, s, n;
    citește n;
    p:=1; s:=0;
    dacă  $n \bmod 5 = 0$  atunci p:=p*n
        altfel s:=s +n
        sfârșit dacă;
    s:=s+n;
    p:=p*n;
    scrie p,s;
stop
```

32) Se citește de la tastatură un număr natural întreg format din 3 cifre. Să se afișeze cel mai mare număr care se poate forma din cifrele sale. Modificați programul astfel încât să afișeze cel mai mic număr ce se poate forma din cifrele numărului citit. Modificați programul astfel încât nu se permită ca numărul să înceapă cu cifra 0.

33) Scrieți un program care să permită alegerea unei opțiuni dintr-un meniu afișat pe ecran, apoi alegeți o operație din meniu prin introducerea numărului de ordine. Meniul conține: 1. ORDONARE CRESCĂTOARE 2. ORDONARE DESCRESCĂTOARE . Programul ordonează crescător și descrescător cifrele unui număr întreg format din 3 cifre.

34) Se citesc patru numere naturale. Să se afișeze cea mai mică și cea mai mare fracție care se poate forma cu două din cele patru numere.

35) Un schior își dorește un traseu care să conțină cel puțin un pisc (vârf) și cel puțin o vale. Dacă traseul este caracterizat (ca altitudini relative) de cifrele componente ale unui număr dat de patru cifre, ajutați-l să stabilească dacă îl parcurge sau nu. Exemplu: traseul 5745 poate fi parcurs, traseele 1234 și 6855 nu pot fi parcurse.

36) Se dă un număr natural n . Să se determine cele mai mari două numere impare, mai mici decât n .

37) Fie n un număr natural, citit. Scrieți un program C++, care să determine ultima cifră a puterii 2^n .

38) Scrieți un program C++ în care se citesc două numere naturale. Să se verifice dacă fracția care se poate forma cu numerele citite este echiunitară, subunitară sau supraunitară.

39) Ionel are voie să se uite la TV 20 de ore pe săptămână. Se introduce numărul de ore în care se uită la TV în fiecare zi din săptămână. Să se verifice dacă va fi pedepsit sau nu. Exemplu: Date de intrare: 3 4 2 2 5 6 1 Date de ieșire: Va fi pedepsit.

40) Numim pereche palindromică două numere x, y citite cu proprietatea că al doilea număr citit de la dreapta la stânga este egal cu primul. Scrieți un algoritm care verifică pentru două numere citite, care au exact 3 cifre fiecare, dacă au sau nu această proprietate.

41) Se dă un număr întreg format din exact patru cifre și se cere să se tipărească: suma cifrelor lui și produsul cifrelor lui. Pentru numerele construite astfel: din cifra miilor și cifra unităților se va construi un număr; din cifra zecilor și cifra sutelor, cel de-al doilea număr; iar dintre cele două, se va tipări cel mai mare.

Ex: pentru numărul 4159 se vor tipări datele: suma= 19, produs=180, cel mai mare dintre 49 și 51 este 51 .

42) Ce se va afișa pe ecran în urma execuției următoarei secvențe de program dacă se citește a) 5 b) 8 c) 15?

```
int x;
cin>>x;
if(x>7)
if(x<11)
cout<<x;
else cout<<x+1;
else cout<<x-1;
```

43) Se citesc două numere întregi a și b . Dacă a divide pe b sau b divide pe a , atunci să se afișeze câtul împărțirii celui mai mare la cel mai mic, iar altfel, restul împărțirii celui mai mare la cel mai mic.

44) Se dau două numere. Să se înmulțească cel mai mare cu 2 și cel mai mic cu 3 și să se afișeze rezultatele. Exemplu : date de intrare : 3 7 date de ieșire : 9 14.

45) Scrieți un algoritm în pseudocod pentru calculul următoarei expresii, pentru x citit:

$$E(x) = \begin{cases} x^3 + 2x & \text{dacă } x \geq 11; \\ x+5 & \text{dacă } 5 \leq x < 11; \\ x-5 & \text{dacă } x < 5; \end{cases}$$

46) Se introduc 2 numere. Să se afișeze un mesaj dacă aceste numere sunt pare consecutive.

Bibliografie

MILOȘESCU Mariana, Informatică, Manual pentru clasa a IX-a, Editura Didactică și Pedagogică R.A.

Bibliografie web:

www.pbinfo.ro

www.campion.ro

www.varena.ro