

Consultații pentru elevii de liceu  
organizate de Facultatea de Matematică și Informatică  
pentru pregătirea concursului Mate-Info și concursului  
de admitere 2024

11.11.2023, 13.00 – 14.15 Drd. Ioan-Daniel POP

Algoritmi care lucrează pe numere  
(fără tablouri sau alte elemente structurate)

Problema 1

Scrieți valorile care se vor afișa pentru  $x = 140$  și  $y = 15$

citește  $x, y$  (numere naturale)

$r \leftarrow x$

cât timp  $y \leq r$  execută

$r \leftarrow r - y$

sfârșit cât timp

$k \leftarrow (x-r)/y$

scrie  $k, r$

**Răspuns:** se vor afișa câtul și restul împărțirii lui  $x$  la  $y$ , adică 9 și 5

Problema 2

Scrieți valorile care se vor afișa pentru  $a = 8$  și  $b = 12$

citește  $a, b$  (numere naturale,  $a \leq b$ )

$y \leftarrow 0$

pentru  $x \leftarrow b, a, -1$  execută

$y \leftarrow x$

$k \leftarrow 0$

```

cât timp  $y > 0$  execută
     $k \leftarrow k + 1 - 2 * (y \% 2)$ 
     $y \leftarrow [y / 2]$ 
sfârșit cât timp
dacă  $k = 0$  atunci
    scrie  $x$ , " "
sfârșit dacă
sfârșit pentru
scrie "*"

```

**Răspuns:** se vor afișa în ordine descrescătoare toate numerele din intervalul  $[a, b]$  cu proprietatea că în reprezentarea în baza 2 numărul de cifre 1 este egal cu numărul de cifre 0; se afișează: 12, 10, 9 \*

### Problema 3

Scrieți valorile care se vor afișa pentru  $n = 5$  și  $x = 7$

citește  $n$ ,  $x$  ( $n$  - număr natural,  $x$  cifră nenulă)

pentru  $i \leftarrow 1, n$  execută

```

     $j \leftarrow 2$ 

```

```

    cât timp  $j \leq [x / j]$  și  $x \% j \neq 0$  execută

```

```

         $j \leftarrow j + 1$ 

```

```

    sfârșit cât timp

```

```

    dacă  $j * j > x$  atunci

```

```

        scrie  $x$ , " "

```

```

    sfârșit dacă

```

```

     $x \leftarrow x + 10$ 

```

```

sfârșit pentru

```

**Răspuns:** se vor afișa numerele prime dintre primele  $n$  numere naturale care se termină cu cifra  $x$ . Se vor afișa 7, 17, 37, 47

## Problema 4

Fiind dat un număr natural, să se calculeze cifra de control a numărului citit. Cifra de control a unui număr se obține efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, până când suma obținută este un număr format dintr-o singură cifră. Această ultimă cifră poartă numele de cifră de control.

Exemplu:  $n = 54377 \Rightarrow$  cifra de control = 8

a) Varianta recursivă

$n = 54377 \Rightarrow$  suma cifrelor lui  $n = 26$ , după prima iterație  $n$  devine  $s \Rightarrow n = 26$  și aplicăm din nou algoritmul, suma cifrelor noului  $n$  este 8  $\Rightarrow$  cifră\_control = 8

Funcția *suma\_cifre* calculează suma cifrelor unui număr întreg.

```
1 #include <iostream>
2 using namespace std;
3 int suma_cifre (int n)
4 {
5
6     int s = 0;
7     while (n != 0)
8     {
9         s += n % 10;
10        n /= 10;
11    }
12    return s;
13 }
14
15 int main ()
16 {
17     int n;
18     cout << "Dați numărul n = ";
19     cin >> n;
20     while (n > 9)
21     {
22         n = suma_cifre (n);
23     }
24     cout <<"Cifra de control a numărului dat este " <<n;
25     return 0;
26 }
```

b) Varianta iterativă

Tot pentru calcularea cifrei de control se poate folosi și următoarea metodă: se trunchiază numărul prin eliminarea ultimei cifre apoi se adună cu aceasta până se obține un număr format dintr-o singură cifră.

Exemplu:

$n = 54377$

$n = 5437 + 7 = 5444$

$$n = 544 + 4 = 548$$

$$n = 54 + 8 = 62$$

$$n = 6 + 2 = 8 \Rightarrow \text{cifra de control} = 8$$

```
1  #include <iostream>
2  using namespace std;
3
4  int main ()
5  {
6      int n;
7      cin >> n;
8      while (n > 9)
9      {
10         n = n / 10 + n % 10;
11     }
12     cout << n;
13 }
```

c) Varianta directă

Pornind de la criteriul de divizibilitate cu 9, un număr se divide cu 9 dacă suma cifrelor sale este multiplu de 9 și cum cifra cea mai mare în baza 10 este 9, putem afla cifra de control calculând restul împărțirii numărului la 9. Dacă restul este 0 atunci cifra de control este 9, altfel este acel rest.

Exemplu:

$$n = 54377 \Rightarrow n \% 9 = 54377 \% 9 = 8$$

```

1  #include <iostream>
2  using namespace std;
3
4  int main ()
5  {
6      int n;
7      cin >> n;
8      if (n % 9 == 0)
9          cout << 9;
10     else
11         cout << n % 9;
12
13     return 0;
14 }

```

### Problema 5

Se citesc două numere naturale  $a$  și  $b$ ,  $a < b$ . Să se afișeze toate perechile de **numere prietene** din intervalul  $[a, b]$ . Două numere se numesc prietene dacă fiecare din ele este egal cu suma divizorilor celuilalt număr, excluzând numărul însuși. De exemplu, numerele 220 și 284 sunt prietene. Divizorii lui 220 sunt:  $1+2+4+5+10+11+20+22+44+55=284$ , iar ai lui 284 sunt :  $1+2+4+71+142=220$ .

Pentru rezolvarea problemei, folosim o funcție auxiliară pentru a calcula suma divizorilor unui număr (există și variante mai eficiente pentru calculul sumei divizorilor).

```

1 #include <iostream>
2 using namespace std;
3
4 int sumaDivizorilor (int numar)
5 {
6     int suma = 1;
7     for (int i = 2; i <= numar / 2; i++)
8     {
9         if (numar % i == 0)
10        {
11            suma += i;
12        }
13    }
14    return suma;
15 }
16
17 int main ()
18 {
19     int a, b;
20     cout << "Introduceti valorile pentru a si b ";
21     cin >> a >> b;
22     if ((sumaDivizorilor (a) == b) && (a == sumaDivizorilor (b)))
23         cout << "Sunt numere prietene";
24     return 0;
25 }

```

## Problema 6

Se citește un număr natural  $n$ . Să se determine cea mai mare cifră a reprezentării lui  $n$  în baza  $p$  și de câte ori se folosește această cifră pentru a-l reprezenta pe  $n$  în baza  $q$ , știind că  $2 \leq p$ ,  $q \leq 9$  și  $p \leq q$ .

Vom diviza problema în mai multe subprograme, pentru lizibilitatea și modularizarea codului.

### Specificații funcției:

*Funcția maxim* - funcție pentru determinarea maximului dintre două numere

*date de intrare:* două numere întregi

*date de ieșire:* numărul maxim dintre cele două

*Funcția ceaMaiMareCifra* - funcție pentru determinarea celei mai mari cifre în reprezentarea numărului număr în baza specificată baza.

*date de intrare:* un număr întreg, o bază de numerație (număr întreg)

*date de ieșire:* cifra maximă a numărului număr în scrierea în baza dată ca input

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int maxim (int a, int b)
6 {
7     return a > b ? a : b;
8 }
9
10 int ceaMaiMareCifra (int numar, int baza)
11 {
12     int maxCifra = 0;
13     while (numar > 0)
14     {
15         int cifra = numar % baza;
16         maxCifra = maxim (maxCifra, cifra);
17         numar /= baza;
18     }
19     return maxCifra;
20 }

```

*Funcția numaraCifraInBazaQ* - funcție pentru determinarea numărului de apariții a celei mai mari cifre în reprezentarea numărului număr în baza specificată baza.

*date de intrate:* un număr întreg, cifra maximă a acestuia, o bază de numerație (număr întreg)

*date de ieșire:* returnează numărul de apariții ale cifrei maxime (cifraMax) în reprezentarea numărului număr în baza specificată baza.

```

22 int numaraCifraInBazaQ (int numar, int cifraMax, int baza)
23 {
24     int count = 0;
25     while (numar > 0)
26     {
27         int cifra = numar % baza;
28         if (cifra == cifraMax)
29         {
30             count++;
31         }
32         numar /= baza;
33     }
34     return count;
35 }
36

```

În funcția main apelăm funcțiile scrie mai sus pentru rezolvarea efectivă a programului.

```

37 int main ()
38 {
39     int n, p, q;
40     cout << "Introduceti valoarea pentru n: ";
41     cin >> n;
42     cout << "Introduceti valoarea pentru p: ";
43     cin >> p;
44     cout << "Introduceti valoarea pentru q: ";
45     cin >> q;
46     int cifraMax = ceaMaiMareCifra (n, p);
47     int aparitii = numaraCifraInBazaQ (n, cifraMax, q);
48     cout << "Cea mai mare cifra in reprezentarea in baza " << p << " este: " <<
49         cifraMax << endl;
50     cout << "Cea mai mare cifra apare de " << aparitii <<
51         " ori in reprezentarea in baza " << q;
52
53     return 0;
54 }

```

## Grile

### Grila 1

Se consideră variabilele de tip întreg  $a = 15$ ,  $b = 5$ ,  $c = 6$  și  $d = 3$  și  $R$ . Indicați care dintre instrucțiunile de mai jos conduce la aceeași valoare pentru  $R$ ?

1.  $R = c / (b * d) * a$  //  $R = 0$
2.  $R = a * c / b * d$  //  $R = 54$
3.  $R = a * c / b / d$  //  $R = 6$
4.  $R = a / b * c / d$  //  $R = 6$

- a) 1 și 4      b) 2 și 3      c) 2 și 4      d) 1 și 2      e) 1 și 3      **f) 3 și 4**

### Grila 2

Variabila  $a$  este de tip real. Pentru a verifica dacă valoarea variabilei  $a$  aparține mulțimii  $[-3, 2] \cup \{3, 5, 9\}$  se va utiliza următoarea expresie:

- a)  **$!((a < -3) \parallel (a > 2)) \parallel (a == 3) \parallel (a == 5) \parallel (a == 9)$**
- b)  $(a >= -3) \&\& (a <= 2) \&\& (a == 3) \parallel (a == 5) \parallel (a == 9)$
- c)  $(a > -3) \&\& (a < 2) \parallel (a == 3) \parallel (a == 5) \parallel (a == 9)$
- d)  $(a < -3) \parallel (a > 2) \&\& (a == 3) \&\& (a == 5) \&\& (a == 9)$
- e)  $(a >= -3) \parallel ((a <= 2) \&\& (a == 3)) \parallel (a == 5) \parallel (a == 9)$
- f)  $(a >= -3) \parallel !((a > 2)) \parallel (a == 3) \parallel (a == 5) \parallel (a == 9)$

### Grila 3

Care din următoarele secvențe de pseudocod afișează DA numai în cazul în care numărul natural  $n \geq 2$  este prim?

(1)  $i \leftarrow 1$   
 $cnt \leftarrow 0$   
cât timp  $i < n$  execută  
    dacă  $n \bmod i = 0$  atunci  
         $cnt \leftarrow cnt + 1$   
     $i \leftarrow i + 1$   
dacă  $cnt = 2$  atunci  
    scrie "DA"

(2)  $i \leftarrow 2$   
 $cnt \leftarrow 0$   
cât timp  $i * i < n$  execută  
    dacă  $n \bmod i = 0$  atunci  
         $cnt \leftarrow cnt + 1$   
     $i \leftarrow i + 1$   
dacă  $cnt = 0$  atunci  
    scrie "DA"

(3)  $i \leftarrow 2$   
 $cnt \leftarrow 1$   
cât timp  $i \leq n$  execută  
    dacă  $n \bmod i = 0$  atunci  
         $cnt \leftarrow cnt + 1$   
     $i \leftarrow i + 1$   
dacă  $cnt = 2$  atunci  
    scrie "DA"

(4)  $i \leftarrow 1$   
 $cnt \leftarrow 0$   
cât timp  $i * i \leq n$  execută  
    dacă  $n \bmod i = 0$  atunci  
         $cnt \leftarrow cnt + 1$   
     $i \leftarrow i + 1$   
dacă  $cnt = 0$  atunci  
    scrie "DA"



a) 1

b) 2

c) 3

d) 4

#### Grila 4

Se consideră următorul algoritm:

```
citește a, b (numere naturale nenule)
pentru k ← 1, a+b execută
  dacă a mod k + b mod k = 0 atunci
    c ← (a div k) * b
scrie c
```

Precizați ce va afișa întotdeauna algoritmul de mai sus:

a)  $(a * b) \text{ div } (a + b)$

b) cel mai mare divizor comun al numerelor **a** și **b**

c) cel mai mare divizor al numărului **a\*b**

d) cel mai mic multiplu comun al numerelor **a** și **b**

#### Grila 5

Precizați ce valoare are variabila de tip întreg **a** după executarea următoarei secvențe de instrucțiuni.

```
a=2019;
for (x=1; x<=5; x++);
a=a+2;
```

a) 2019

b) 2020

c) 2021

d) 2024

e) 2027

f) 2029

**Explicații:** instrucțiunea *for* nu produce niciun efect

#### Grila 6

Variabila **n** reprezintă un număr natural cu exact două cifre. Precizați care dintre expresiile următoare au valoarea **1/true** dacă și numai dacă cifrele lui **n** au aceeași paritate.

1.  $(n/10 - n\%10) \% 2 == 0$
2.  $n/10 \% 2 == n\%2$
3.  $n/10 == n\%10$
4.  $(n/10 + n\%10 * 10) \% 2 == n\%2$
5.  $n/2 == n\%2$

b) 1   c) 2   d) 3   e) 4   f) 5

### Grila 7

Se consideră următorul program:

```
#include<iostream>
using namespace std;
int main( )
{
    int n, cn, x=0,p=1;
    cin>>n;
    cn=n;
    while(n)
    {
        if (n%10>x) x=n%10;
        n/=10;
    }
    x++;
    while(cn)
    {
        n=n+cn%10*p;
        p*=x;
        cn/=10;
    }
    cout<<n;
}
```

Precizați care este cel mai mic număr natural format din 5 cifre distincte care poate fi citit ca dată de intrare astfel încât valoarea afișată să fie aceeași.

- a) 10000   b) 10162   c) 10234   d) 10239   e) 10723   f) 12345

**Explicații:** algoritmul determină baza minimă  $x$  în care îl consideră pe  $n$ -ul inițial și îl transformă în baza 10. Varianta a (10000) poate fi scrisă în baza minimă 2, b poate fi scris în baza minimă 7, c în baza minimă 5, d în baza minimă 10, e în baza minimă 8, iar f în baza minimă 6. Baza minimă în care putem reprezenta un număr este dată de formula cifra maximă + 1. Deoarece ne dorim ca valoarea afișată să fie egală cu valoarea inițială, înseamnă că baza minimă trebuie să fie 10. Doar varianta d are baza minimă de reprezentare egală cu 10.

### Grila 8

Dacă pentru variabila **a** se citește valoarea **11**, precizați care valori pot fi citite pentru variabila **b** astfel încât în urma apelului **f(a,b)**, subprogramul **f** definit mai jos să producă valoarea **10**.

```
int f(int a,int b)
{  if(a==b)
    return a%2;
   else
   return
f(a, (a+b)/2)+f((a+b)/2+1,b); }
```

a) 29   b) 19   c) 31   d) 30   e) 28   f) 25

**Explicații:** subprogramul calculează numărul valorilor naturale impare din intervalul [a, b]. Ne dorim să afișăm valoarea 10, ceea ce înseamnă că în intervalul [11, b] trebuie să fie 10 valori impare  $\Rightarrow b = 29$  sau  $b = 30$

### Grila 9

Precizați care este valoarea produsă de subprogramul **f**, definit mai jos, în urma apelului **f(1999,7)**.

```
int f̄(int x, int n)
{if(n)
if(n%2==0)
return (f(x,n/2)*f(x,n/2))%10;
else
return (x*f(x,n-1))%10;
else
return 1;}
```

a) 798800599919   b) 9   c) nu poate fi calculată valoarea   d) 1   e) 0   f) 28

**Explicații:** subprogramul calculează ultima cifră a lui  $x$  la putere  $n$ .

### Grila 10

Fie secvența de cod următoare, în care se consideră că variabilele **a, i, n** rețin numere întregi:

```

cin>>n; a = 1; i = 2;
while(i<n && a>0){
    if(n%i == 0) a=0;
    else
        i++; cout<<i;}

```

Definim, în acest context, **operație** drept o instrucțiune de *atribuire* sau o expresie de *incrementare*. Care este numărul **minim** de **operații** ce se pot executa în secvența de mai sus, în funcție de valoarea citită pentru variabila **n**?

- a)  $2n-1$       b)  $n-1$       c) 5      d) 3      **e) 2**      f) 4

**Explicații:** pentru cazul când  $i$  este mai mare decât  $n$ , se vor face doar atribuiri de pe prima linie.

### Grila 11

Fie secvența de cod următoare în care se consideră că variabilele **a, i, n** rețin numere întregi.

```

int main(){
    cin>>n; a = 1; i = 2;
    while (i<n && a>0) {
        if(n%i == 0) a = 0;
        else i++; cout<<i;
    }
}

```

Definim, în acest context, **operație** drept o instrucțiune de *atribuire* sau o expresie de *incrementare*. Care este numărul **maxim** de **operații** ce se pot executa în secvența de mai sus, în funcție de valoarea citită pentru variabila **n**?

- a)  $2n+2$       b)  $2n$       c)  $n-1$       d)  $2n+3$       **e) n**      f)  $n+1$

**Explicații:** cazul cel mai defavorabil este atunci când  $n$  este prim. În acest caz, înainte de structura while se fac 2 atribuiri, iar în cadrul structurii repetitive sunt  $n - 2$  instrucțiuni de atribuire. În total  $n$  instrucțiuni de atribuire.

### Grila 12

Subprogramul **f** este definit mai jos. O condiție necesară și suficientă pentru ca numărul natural mai mare strict ca **1** reținut de variabila **n** să fie prim este:

```

int f(int d, int n)
{
    do
    {
        d++;
    }
    while (n % d != 0);
    return d;
}

```

- a)  $f(2, n) == n$                       b)  $f(2, n) == 2$   
c)  $f(1, n) == n$                       d)  $f(1, n) == 1$   
e)  $f(1, n - 1) == n$                   f)  $f(2, n-1) == 2$

**Explicații:** apelul  $f(d, n)$  returnează cel mai mic divizor al lui  $n$  mai mare strict decât  $d$ . În cazul în care  $n$  este prim acesta va fi  $n$  și reciproc (în enunț se garantează că  $n$  este natural, mai mare strict ca 1)

### Grila 13

Expresia corespunzătoare mediei aritmetice a patru numere reale memorate în variabilele **a**, **b**, **c** și **d** este:

- a)  $(a+b+c+d)/4$                       b)  $(a+b+c+d)*1/2$   
c)  $(a+b+c+d)*0.4$                       d)  $(a+b+c+d)*0.25$   
e)  $(a+b+c+d)*4.0$                       f)  $(a+b+c+d)*1.4$

**Explicații:** expresiile de la variantele b, c, e și f nu sunt corecte din punct de vedere matematic.

### Grila 14

În secvențele de instrucțiuni **S1** și **S2** variabilele **n** și **p** sunt de tip întreg. Obținerea în variabila **p** a primei cifre a numărului reținut inițial de **n** este realizată:

```

//S1
p = n;
while (p > 9)
{
    p /= 10;
}

//S2
do
{
    p = n % 10;
    n /= 10;
}
while (n != 0);

```

- a) doar de **S1**  
b) doar de **S2**  
c) atât de **S1**, cât și de **S2**

- d) doar folosind o cu totul altă secvență
- e) doar de **S1**, dacă **n** are o singură cifră
- f) doar de **S2**, dacă **n** are mai multe cifre

**Explicații:** ambele secvențe de instrucțiuni au ca efect obținerea în variabila  $p$  a primei cifre a numărului reținut inițial de  $n$ . În cadrul **S1**  $p$  reține mai întâi o copie a lui  $n$ , iar apoi, cât timp  $p$  are mai mult de o cifră se elimină ultima dintre acestea. În **S2** se reține în mod repetat în  $p$  ultima cifră a lui  $n$ , apoi aceasta se elimină prin păstrarea în  $n$  a câtului împărțirii întregi a lui  $n$  la 10. În felul acesta ultima cifră eliminată este evident prima cifră a numărului reținut inițial de  $n$ , aceasta fiind valoarea finală a lui  $p$ . Singura diferență dintre **S1** și **S2** este că în cazul celei de-a doua secvențe de instrucțiuni valoarea inițială a lui  $n$  se va pierde, aceasta fiind înlocuită cu 0 la sfârșitul executării buclei.

### Grila 15

**Variabilele  $a$ ,  $b$ ,  $c$  memorează numere naturale nenule. Instrucțiunea  $c=b-b\%a$  atribuie variabilei  $c$  o valoare care reprezintă:**

- a) cel mai mic număr natural mai mare sau egal cu  $a$  și care este divizibil cu  $b$ ;
- b) cel mai mic număr natural mai mare sau egal cu  $b$  și care este divizibil cu  $a$ ;
- c) cel mai mic număr natural mai mare sau egal cu  $b$  și care este nedivizibil cu  $a$ ;
- d) cel mai mare număr natural mai mic sau egal cu  $b$  și care este divizibil cu  $a$ ;**
- e) cel mai mare număr natural mai mic sau egal cu  $a$  și care este divizibil cu  $b$ ;
- f) cel mai mare număr natural mai mic sau egal cu  $a$  și care este nedivizibil cu  $b$ .