

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Algoritmi care lucrează pe numere

### Problema 1

Scrieti un program care sa determine al n-lea numar din sirul lui Fibonacci.

#### Exemple:

Pentru  $n=3$  => se va afisa 2

Pentru  $n=5$  => se va afisa 5

Pentru  $n=7$  => se va afisa 13

Pentru  $n=9$  => se va afisa 34

Pentru  $n=10$  => se va afisa 55

Pentru  $n=11$  => se va afisa 89

Varianta 1 de rezolvare – *varianta recursiva – implementare C++*

```
#include <iostream>

/* crearea functiei pentru determinarea celui al n-lea numar din sirul lui
Fibonacci
    Date de intrare: numarul n
    Date de iesire: numarul din sir corespunzator
*/

int functiel(int n)
{
    if (n <= 1)
        return n;
    return functiel(n-1) + functiel(n-2);
}

int main()
{
    int n;
    std::cout <<"introduceti numarul n=";
    std::cin >>n;

    std::cout <<"Rezultat= "<<functiel(n)<<std::endl;
    return 0;
}
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

Varianta 1 de rezolvare – *varianta recursiva – implementare Pascal*

```
program problema1;
var numar:integer;
{ crearea functiei pentru determinarea celui al n-lea numar din sirul lui
Fibonacci
    Date de intrare: numarul n
    Date de iesire: numarul din sir corespunzator
}

function functie1(n: integer ):integer;
begin
    if (n <= 1) then functie1 := n
    else functie1 := functie1(n-1) + functie1(n-2);

end;
begin
write('Introduceti numarul n=');
readln(numar);
writeln( 'Rezultatul este: ', functie1(numar));
end.
```

Varianta 2 de rezolvare – *varianta iterativa – implementare C++*

```
#include <iostream>

/* crearea functiei pentru determinarea celui al n-lea numar din sirul lui
Fibonacci
    Date de intrare: numarul n
    Date de iesire: numarul din sir corespunzator
*/

int functie2(int n)
{
    int a = 0, b = 1, c, i;
    if( n == 0)
        return a;
    for (i = 2; i <= n; i++)
    {
        c = a + b;
        a = b;
        b = c;
    }
    return b;
}
```

**Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică**  
**Consultații la Informatică pentru pregătirea concursului de admitere 2020**

---

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
int main()
{
    int n;
    std::cout <<"introduceti numarul n=";
    std::cin >>n;

    std::cout <<"Rezultat= "<<functie2(n)<<std::endl;
    return 0;
}
```

Varianta 2 de rezolvare – *varianta iterativa – implementare Pascal*

```
program problemaliterativ;
var numar:integer;

{ crearea functiei pentru determinarea celui al n-lea numar din sirul lui
Fibonacci
  Date de intrare: numarul n
  Date de iesire: numarul din sir corespunzator
}

function functie2(n: integer ):integer;
var a,b,c,i:integer;
begin
    a:=0;
    b:=1;

    if( n = 0) then functie2 :=a
    else

    for i:= 2 to n do
    begin
        c := a + b;
        a := b;
        b := c;
    end;
    functie2:=b;
end;

begin
write('Introduceti numarul n=');
readln(numar);
writeln( 'Rezultatul este: ', functie2(numar));
end.
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Problema 2

Sa se determine toate numerele palindroame mai mici decat un numar  $n$  dat.

### Exemple:

Pentru  $n = 12 \Rightarrow$  se va afisa: 1, 2, 3, 4, 5, 6, 7, 8, 9, 11

Pentru  $n = 104 \Rightarrow$  se va afisa: 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101

### Rezolvare – Implementare C++

```
#include <iostream>
using namespace std;

// functie pentru crearea unui palindrom
int crearePalindrom(int input, int b, bool esteImpar)
{
    int n = input;
    int palindrom = input;

    // verificam daca numar cifrelor este par sau impar
    // daca numarul cifrelor este impar, atunci
    // nu tinem cont de ultima cifra de intrare in
    // identificarea inversului

    if (esteImpar)
        n /= b;

    // crearea efectiva a palindromului pe baza formulei
    while (n > 0)
    {
        palindrom = palindrom * b + (n % b);
        n /= b;
    }
    return palindrom;
}
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
// functie pentru generarea tuturor palindroamelor si afisarea lor
int generarePalindroame(int n)
{
    int number;

    // de doua ori pentru palindroame de lungime para si impara
    for (int j = 0; j < 2; j++)
    {
        int i = 1;
        while ((number = crearePalindrom(i, 10, j % 2)) < n)
        {
            cout << number << " ";
            i++;
        }
    }
    return 0;
}

int main()
{
    int n;
    std::cout << "introduceti numarul n=";
    std::cin >> n;

    std::cout << generarePalindroame(n) << std::endl;
    return 0;
}
```

## Rezolvare – Implementare Pascal

```
{functie pentru a testa daca oglinditul/inversul unui numar este egal cu
numarul}
function test_palindrom(numar:longint):boolean;
begin
    if (numar = oglindit(numar)) then test_palindrom := true
    else test_palindrom := false;
end;

{programul principal al problemei}
begin
write('Introduceti numarul n=');
readln(n);
    for i := 0 to n do
        if (test_palindrom(i)) then
            write(i, ' ');
end.
```

11 ianuarie 2020

Lect. Dr. Mihai Suciuc  
Asist. Dr. Adriana Coroiu

```
program palindrom;
var n,i:integer;
{functie pentru crearea oglinditului/inversului unui numar}
function oglindit(numar_original:longint):longint;
var aux, numar_oglindit: longint;
begin
  numar_oglindit := 0;
  while (numar_original <> 0) do
  begin
    numar_oglindit := numar_oglindit * 10 + numar_original mod 10;
    numar_original := numar_original div 10;
  end;
  oglindit := numar_oglindit;
end;
```

### Problema 3

Se da ecuatia de gradul 2 de forma:  $X^2 - s * X + p = 0$ , cu  $s, p$  apartinand lui  $R$  si  $n$  apartinand lui  $N$ .

Sa se calculeze, in mod recursiv suma puterilor radacinilor  $X_1^n + X_2^n$ , fara a se calcula radacinile ecuatiei  $X_1$  si  $X_2$ .

#### Exemple:

Daca avem ecuatia:  $X^2 - 6 * X + 8 = 0$  si  $n=2 \Rightarrow$  rezultatul este 20 ( $x_1 = 4$  si  $x_2 = 2$ )

Daca avem ecuatia:  $X^2 - 8 * X + 15 = 0$  si  $n=2 \Rightarrow$  rezultatul este 34 ( $x_1 = -3$  si  $x_2 = -5$ )

#### Explicatii:

Stiind ca  $X_1$  si  $X_2$  sunt radacinile ecuatiei, rezulta relatiile:

$$X_1^2 - s * X_1 + p = 0$$

$$X_2^2 - s * X_2 + p = 0$$

Daca vom inmulti prima relatie cu  $X_1^n$  si pe cea de a doua cu  $X_2^n$  vom obtine:

$$X_1^{n+2} - s * X_1^{n+1} + p * X_1^n = 0$$

$$X_2^{n+2} - s * X_2^{n+1} + p * X_2^n = 0$$

Daca insumam cele doua relatii, vom obtine:

$$(X_1^{n+2} + X_2^{n+2}) - s (X_1^{n+1} + X_2^{n+1}) + p (X_1^n + X_2^n) = 0$$

Ceea ce am putea scrie in mod echivalent:  $S_{n+2} - s * S_{n+1} + p * S_n = 0$  rezulta deci:  $S_{n+2} = s * S_{n+1} - p * S_n$

Vom obtine de aici urmatoarea relatie de recurenta pentru suma puterilor radacinilor unei ecuatie de gradul II:

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

$$Sum(n) = \begin{cases} 2, & \text{daca } n = 0 \\ s, & \text{daca } n = 1 \\ s * Sum(n-1) - p * Sum(n-2), & \text{daca } n > 1 \end{cases}$$

## Rezolvare – implementare C++

```
#include <iostream>
float s,p;

/*crearea functiei pentru calculul sumei conform formulei de recurenta
   Date de intrare: numarul n
   Date de iesire: rezultatul sumei conform formulei de recurenta
*/

float Sum (int n)
{
    if (!n) return 2;
    if (n==1) return s;
    return s*Sum(n-1)-p*Sum(n-2);
}
int main()
{
    int n;
    std::cout <<"introduceti cei doi coeficienti: ";
    std::cin >>s>>p;
    std::cout << "n=";
    std::cin >>n;
    std::cout <<"Rezultatul este: "<<Sum(n)<<std::endl;
    return 0;
}
```

## Rezolvare – implementare Pascal

```
program problema3;
var numar,s,p:integer;

{crearea functiei pentru calculul sumei conform formulei de recurenta
  Date de intrare: numarul n
  Date de iesire: rezultatul sumei conform formulei de recurenta
}
function suma(n: integer ):integer;
begin
    if n<=0 then suma:=2
    else
        if n=1 then suma:=s
        else suma:=s*suma(n-1)-p*suma(n-2);
end;
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
begin
write('Introduceti numarul n=');
readln(numar);

write('Introduceti coeficientul s=');
readln(s);

write('Introduceti coeficientul p=');
readln(p);

writeln('Rezultatul este: ', suma(numar));
end.
```

## Problema 4

a) Sa se scrie un subprogram care prin intermediul a trei parametri, notați a, b și c, reprezentand trei valori naturale nenule, fiecare de maximum patru cifre va returna valoarea 1 dacă cele trei valori ale celor trei parametri pot constitui laturile unui triunghi și valoarea 0 în caz contrar.

### Varianta C++

```
int verificare (int a, int b, int c)
{
    return ((a + b > c) && (a + c > b) && (b + c > a));
}
```

### Varianta Pascal

```
function verificare(a,b,c:integer):integer;
begin
    if(a+b>c) and (a+c>b) and (b+c>a)
        then verificare:=1
        else verificare:=0;
end;
```

b) Sa se scrie un program care citește de la tastatură șase valori naturale nenule, apoi verifică, utilizând apeluri utile ale subprogramului de la subpunctul anterior dacă primele trei numere citite pot constitui laturile unui triunghi și dacă ultimele trei numere citite pot constitui laturile unui triunghi.

In caz afirmativ, programul va afisa pe ecran mesajul **congruente** dacă cele două triunghiuri sunt congruente sau mesajul **necongruente** dacă cele două triunghiuri nu sunt congruente.

Dacă cel puțin unul dintre cele două triplete de valori nu pot constitui laturile unui triunghi, programul va afișa pe ecran mesajul **nu se poate forma un triunghi**.



11 ianuarie 2020

Lect. Dr. Mihai Suciu  
Asist. Dr. Adriana Coroiu

**Obs.** Triunghiurile sunt congruente dacă există cele două triunghiuri și dacă au laturile corespunzătoare congruente (cazul L.L.L.).

Deci, pentru a verifica vom compara lungimile laturilor în ordine corespunzătoare (cea mai mică din primul triunghi cu cea mai mică din al doilea triunghi, până se verifică toate laturile).

### Varianta C++

```
#include <iostream>
using namespace std;
/*
Descriere: Returneaza valoarea 1 daca valorile parametrilor pot forma un
triunghi, 0 in caz contrar.
Date: a, b, c - numare naturale
Rezultate: 1 sau 0
*/
int a, b, c, d, e, f;
int verificare (int a, int b, int c)
{
    return ((a + b > c) && (a + c > b) && (b + c > a));
}

/*
Descriere: Se interschimba valorile celor doua variabile transmise ca si
parametrii
Date: x, y - numare intregi
Rezultate: valorile interschimbate ale parametrilor initiali
*/
void interschimb (int &x, int &y)
{
    int aux = x;
    x = y;
    y = aux;
}

/*
Descriere: Sortarea a trei valori transmise prin parametrii
Date: x, y, z - numare intregi
Rezultate: valorile parametrilor ordonate
*/
void ordonare (int &x, int &y, int &z)
{
    if (x > y) interschimb (x, y);
    if (x > z) interschimb (x, z);
    if (y > z) interschimb (y, z);
}

int main ()
{
    cin >> a >> b >> c >> d >> e >> f;
    if (verificare (a, b, c) && verificare (d, e, f))
```

11 ianuarie 2020

Lect. Dr. Mihai Suciu  
Asist. Dr. Adriana Coroiu

```
{
    ordonare (a, b, c);
    ordonare (d, e, f);
    if (a == d && b == e && c == f)
        cout << "triunghiuri formate congruente";
    else
        cout << "triunghiuri formate necongruente";
}
else
    cout << "NU se poate forma un triunghi";
}
```

### Varianta Pascal

```
var a,b,c,d,e,f:integer;
{
    Descriere: Returneaza valoarea 1 daca valorile parametrilor pot forma un
    triunghi, 0 in caz contrar.
    Date: a, b, c - numare naturale
    Rezultate: 1 sau 0
}
function verificare(a,b,c:integer):integer;
begin
    if(a+b>c) and (a+c>b) and (b+c>a)
        then verificare:=1
        else verificare:=0;
end;

{
    Descriere: Se interschimba valorile celor doua variabile transmise ca si
    parametrul
    Date: x, y - numare intregi
    Rezultate: valorile interschimbate ale parametrilor initiali
}
procedure interschimb(var x,y:integer);
var aux:integer;
begin
    aux:=x;
    x:=y;
    y:=aux;
end;

{
    Descriere: Sortarea a trei valori transmise prin parametrul
    Date: x, y, z - numare intregi
    Rezultate: valorile parametrilor ordonate
}
procedure ordonare(var x,y,z:integer);
begin
    if(x>y) then interschimb(x,y);
    if(x>z) then interschimb(x,z);
    if(y>z) then interschimb(y,z);
end;
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
end;

begin
  readln(a,b,c,d,e,f);
  if(verificare(a,b,c)=1) and (verificare(d,e,f)=1)
    then
      begin
        ordonare(a,b,c);
        ordonare(d,e,f);
        if(a=d) and (b=e) and (c=f)
          then
            writeln('triunghiuri formate congruente')
          else
            writeln('triunghiuri formate necongruente');
        end
      else
        writeln('NU se poate forma un triunghi');
    end.
end.
```

## Problema 5

a) Sa se scrie un subprogram care pentru un parametru  $a$ , numar natural din intervalul  $[2,10000]$ , returneaza **cel mai mic divizor al numarului  $a$** , strict mai mare decat 1.

### Varianta C++

```
int pprim(int a)
{
  int i=2;
  while (a%i)
    i++;
  return i;
}
```

### Varianta PASCAL

```
function pprim (a:integer):integer;
  var i:integer;

begin i:= 2;
  while (a mod i <> 0) do
    i:= i + 1;
  pprim:= i;
end;
```

b) Sa se scrie un program care citeste  $n$  numere naturale (cele  $n$  numere sunt in intervalul  $[2,10000]$  iar  $n$  este in interval  $[1,100]$ ) si afiseaza **cel mai mare numar „aproape prim”** dintre numerele citite, folosind apeluri ale subprogramului de la punctul a).

Un număr natural  $n$  se numește „**aproape prim**” dacă este egal cu produsul a două numere prime distincte.

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

Exemplu: dacă se citesc  $n = 6$  și apoi numerele:

**100 14 21 8 77 35**

atunci se afișează pe ecran **77** pentru că numărul **77** este cel mai mare dintre numerele „aproape prime” citite ( $14=7*2, 21=7*3, 77=7*11, 35=7*5$ ).

*Obs. Un număr „aproape prim” are proprietatea că împărțit la primul său divizor obținem număr prim. Pentru numerele prime, funcția **pprim** returnează valoarea transmisă ca parametru.*

*Nu luăm în considerare numerele care sunt prime.*

### **Varianta C++**

```
#include<iostream>
#include<math.h>
using namespace std;
/*
Descriere: citirea unui numar
Date: n - numar intreg
Rezultate: salvarea valorii introduse in variabila
*/

void citire (int &n)
{
    cout << "n=";
    cin >> n;
}

/*
Descriere: determinarea celui mai mic divizor al numarului a
Date: a - numar intreg
Rezultate: cel mai mic divizor
*/
int pprim (int a)
{
    int i=2;
    while (a%i)
        i++;
    return i;
}

/*
Descriere: determinarea celui mai mare numar aproape prim
Date: n - numar intreg
Rezultate: cel mai mare nr aproape prim
*/
void prelucrare (int n)
{
    int i,a,max=0,p;
    for(i=1;i<=n;i++)
    {
        cin>>a;
        if (pprim(a)!=a)
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
        {
            p=pprim(a);
            if (pprim(a/p)==a/p)
                if (a>max)
                    max=a;
        }
    }
    cout<<max;
}

int main ()
{
    int n;
    citire (n);
    prelucrare (n);
}
```

### Varianta PASCAL

```
{
Descriere: citirea unui numar
Date: n - numar intreg
Rezultate: salvarea valorii introduse in variabila
}
procedure citire (var n:integer);
begin
    write ('n=');
    readln (n);
end;

{
Descriere: determinarea celui mai mic divizor al numarului a
Date: a - numar intreg
Rezultate: cel mai mic divizor
}
function pprim (a:integer):integer;

    var i:integer;

begin i:= 2;
    while (a mod i <> 0) do
        i:= i + 1;
    pprim:= i;
end;

{
Descriere: determinarea celui mai mare numar aproape prim
Date: n - numar intreg
Rezultate: cel mai mare nr aproape prim
}
procedure prelucrare (n:integer);
var i, a, max, p:integer;
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
begin
max:=0;
for i := 1 to n
do
begin
read (a);
if pprim (a) <> a then
begin
p:= pprim (a);
if pprim (a div p) = a div p then
if a > max then max:= a;
end;
end;
write (max);

end;

var n:integer;
begin
citire (n);
prelucrare (n);
end.
```

## Problema 6

Sa se scrie definiția completă a subprogramului **numar**, cu exact doi parametri, care primește prin intermediul parametrului  $x$  un număr natural de cel mult 2 cifre, și prin intermediul parametrului  $y$  un număr natural nenul de cel mult 9 cifre.

Subprogramul va returna cel mai mare număr natural  $z$  pentru care există un număr natural  $k$  astfel încât  $z=x^k$  și  $z<y$ .

**Exemplu:** pentru  $y=18$  și  $x=2$  subprogramul va returna valoarea 16 ( $=2^4<18$ ).

Obs. Esența acestei probleme constă în rezolvarea următoarei inecuații de clasa a X-a:  $x^k < y$  și alegerea valorii naturale maxime a lui  $x$ . Ecuția este echivalentă cu  $k < \log_x y$  și de aici  $k$  maxim este egal cu partea întreagă a logaritmului.

Pentru situațiile în care  $x$  este 0 sau 1, precum și multe altele, problema nu are sens. Aceste cazuri sunt prinse în cadrul funcției **numar**, caz în care se returneaza valoarea -1 (cu sens de eroare).

## Varianta C++

```
#include <iostream>
#include <math.h>

using namespace std;
/*
Descriere: determinarea celui mai mare numar mai mare număr natural care
respecta inegalitatea
Date: n, y - numere intregi
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

Rezultate: cel mai mare număr natural care respecta inegalitatea

```
*/  
long numar (int x, long y)  
{  
  
    //daca x =1 si y=0 sau 1 problema nu are solutie  
    //daca x>=1 si y=0 problema nu are solutie  
    //daca x=y=0 problema deasemenea nu are sens  
    if (x == 0)  
        if (y >= 1)  
            return 0;  
        else  
            return -1;           //eroare  
    else  
        if (x == 1)  
            if (y >= 2)  
                return 1;  
            else  
                return -1;       //eroare  
        else  
            if (y == 0)  
                return -1;       //eroare  
            else  
                return long (log (y) / log (x));  
}  
  
int main ()  
{  
    long x, y;  
    cin >> x;  
    cin >> y;  
    cout << numar (x, y);  
}
```

### **Varianta PASCAL**

```
{  
Descriere: determinarea celui mai mare număr natural care  
respecta inegalitatea  
Date: n, y - numere întregi  
Rezultate: cel mai mare număr natural care respecta inegalitatea  
}  
function numar (x:integer; y: longint): longint;  
var rez:integer;  
begin  
{  
    daca x = 1 si y = 0 sau 1 problema nu are solutie  
    daca x >= 1 si y = 0 problema nu are  
    solutie daca x = y = 0 problema deasemeni nu are sens  
}  
if (x = 0)
```

11 ianuarie 2020

Lect. Dr. Mihai Suciuc  
Asist. Dr. Adriana Coroiu

```
    then if (y >= 1)
    then rez:= 0
        else
            rez:= -1 { eroare }
    else if (x = 1)
        then if (y >= 2)
            then rez:= 1
        else
            rez:= -1 { eroare }
            else if (y = 0)
            then rez:= -1 { eroare }
            else
                rez:= trunc (ln (y) / ln (x));
numar:= rez;
end;
var x, y:integer;
begin
read (x);
read (y);
write (numar (x, y));
end.
```

## Problema 7

Fie  $M$  o multime nevida și  $*$  :  $M \times M \rightarrow M$  o operație binară asociativă definită pe mulțimea  $M$ .

Să se scrie o funcție care să determine numărul de modalități în care se poate evalua expresia  $x_1 * x_2 * \dots * x_n$ , unde  $x_1, x_2, \dots, x_n$  aparțin mulțimii  $M$  și  $n > 0$ .

### Exemple:

Pentru  $n=3$ , numărul de modalități în care se poate evalua expresia  $x_1 * x_2 * x_3$  este 2:  $x_1 * (x_2 * x_3)$  și  $(x_1 * x_2) * x_3$

Pentru  $n=4$ , numărul de modalități de a evalua expresia este 5:  $((x_1 * x_2) * x_3) * x_4$  sau  $(x_1 * x_2) * (x_3 * x_4)$  sau  $(x_1 * (x_2 * x_3)) * x_4$  sau  $x_1 * ((x_2 * x_3) * x_4)$  sau  $x_1 * (x_2 * (x_3 * x_4))$

### Explicații:

În primul rând, pentru  $n=1$  sau  $n=2$  nu avem decât o singură posibilitate (cazurile elementare ale problemei).

Dacă  $n > 2$ , pentru a evalua produsul  $x_1 * x_2 * \dots * x_n$  trebuie să aplicăm asociativitatea și să împărțim termenii în grupându-i între paranteze  $(x_1 * x_2 * \dots * x_k) * (x_{k+1}, x_{k+2}, \dots, x_n)$ .

Vom urma apoi evaluarea parantezelor, urmând să efectuăm operația  $*$  între rezultatele obținute în urma evaluării parantezelor.

Notăm cu  $P(n)$  numărul de posibilități de a evalua o expresie cu  $n$  termeni.



11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

Daca impartim termenii la pozitia  $k$  ( $1 \leq k < n$ ) atunci deducem ca:  $P(n) = P(k) \times P(n-k)$  (pentru orice modalitate de evaluare a primei grupe de  $k$  termeni se poate asocia o modalitate de evaluare a celei de a doua grupe de  $n-k$  termeni).

Impartirea se poate face in orice pozitie  $k$ , incepand cu pozitia 1 pana la pozitia  $n-1$ , se poate deduce urmatoarea formula recursiva pentru calculul  $P(n)$ .

$$P(n) = \begin{cases} 1, & \text{daca } n \leq 2 \\ \sum_{k=1}^{n-1} P(k) * P(n-k), & \text{daca } n > 2 \end{cases}$$

## Rezolvare – implementare C++

```
#include <iostream>
using namespace std;
/* functie pentru efectuarea calculului conform formulei de recurenta de mai
sus
    Date de intrare: numarul n
    Date de iesire: numarul total de posibilitati
*/
unsigned long calcul (int n)
{
    long s=0;
    int k;
    if (n<=2) return 1;
    for (k=1; k<n; k++)
        s+= calcul(k) * calcul(n-k);
return s;
}

int main()
{
    int n;
    std::cout <<"introduceti numarul n=";
    std::cin >>n;

    std::cout<<"Rezultatul este:"<< calcul(n)<<std::endl;
    return 0;
}
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Rezolvare - implementarea in Pascal:

```
program problema4;
var numar:integer;
{functie pentru efectuarea calcului conform formulei de recurenta de mai
sus
    Date de intrare: numarul n
    Date de iesire: numarul total de posibilitati
}
function calcul (n:integer):integer;
var s,k:integer;
begin
    s:=0;
    if n<=2 then calcul:=1
    else
        for k:=1 to n-1 do
            calcul := (s + calcul(k) * calcul(n-k));
end;

begin
write('Introduceti numarul n=');
readln(numar);
writeln('Rezultatul este: ', calcul(numar));
end.
```

## Problema 8

Semafoarele din trei intersecții diferite din centrul orașului își schimbă culoarea după fiecare 48, 72 și 108 secunde. Dacă cele trei semafoare își schimbă simultan culoarea la ora 8 dimineața, la ce altă oră își vor mai schimba simultan culoarea?

### Analiză

- cel mai mic multiplu comun

### Rezolvare problema – implementare C++

```
/*
Problema a fost compilata cu Microsoft Visual Studio varianta Community 2017
*/
#include <iostream>

/*
Descriere:    calculeaza cel mai mic multiplu comun a doua numere naturale
Date:        a, b - numere naturale
Rezultat:    cel mai mic multiplu comun al celor doua numere
*/
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
int cmmmc(int a, int b)
{
    int r;
    if (a < b)
    {
        r = a;
        a = b;
        b = r;
    }
    r = a;
    while ((r%b) != 0)
    {
        r += a;
    }
    return r;
}

int main()
{
    int a = 48;
    int b = 72;
    int c = 108;
    std::cout << "numarul de secunde dupa ora 8 la care semafoarele se vor sincroniza:
" << cmmmc(cmmmc(a, b), c) << std::endl;
    return 0;
}
```

## Rezolvare problema – implementare Pascal

```
program semafoare;
var a,b,c:integer;

{functie care calculeaza cel mai mic multiplu comun al doua numere;
 date de intrare: a, b
 date de iesire: cmmmc (a,b)
}

function cmmmc (a:integer;b:integer):integer;
var x,y,r:integer;
begin
x:=a;
y:=b;
repeat
r:=a mod b;
a:=b;
b:=r
until r=0;
cmmmc:=(x*y) div a;
end;
```

11 ianuarie 2020

Lect. Dr. Mihai Suciu  
Asist. Dr. Adriana Coroiu

```
begin
write('Introduceti numarul a=');
readln(a);
write('Introduceti numarul b=');
readln(b);
write('Introduceti numarul c=');
readln(c);

writeln('Rezultatul este: ', cmmmc(a,cmmmc(b,c)));
end.
```

## Problema 9

Alin are în față  $M$  mingi de tenis roșii și  $N$  mingi de tenis galbene. El dorește să formeze mai multe grupuri, astfel încât toate grupurile să conțină același număr de mingi, toate mingile dintr-un grup să aibă aceeași culoare și numărul de mingi dintr-un grup să fie cât mai mare posibil. După ce a reușit să facă împărțea, Alin a primit alt set de mingi roșii și galbene. Imediat a început să le împartă în grămezi de  $M$  mingi roșii și  $N$  mingi galbene. Fiecare grămadă conține mingi de aceeași culoare. El dorește să aleagă un număr de  $X$  grămezi cu mingi roșii și  $Y$  grămezi cu mingi galbene astfel încât diferența dintre numărul total de mingi roșii și numărul total de mingi galbene să fie egală cu numărul de mingi dintr-unul din grupurile construite la început.

Să se citească de la tastatură numerele  $M$  și  $N$ . Se vor afișa pe ecran numărul de mingi dintr-o grupă construită, numărul de grămezi de mingi roșii și galbene.

## Exemple

Date de intrare	Rezultat
4 6	2 1 1
10 5	5 0 1
24 63	3 8 3

## Analiză

- trebuie determinat cel mai mare divizor comun a două numere  $M$  și  $N$
- trebuie să se determine două numere  $X$  și  $Y$  astfel încât  $|MX - NY| = \text{cmmdc}(M,N)$
- algoritmul extins al lui Euclid

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Rezolvare C++

```
/*
Problema a fost compilata cu Microsoft Visual Studio varianta Community 2017
*/
#include <iostream>

/*
Descriere:   algoritmul lui Euclid in varianta extinsa
Date:       a, b - numere naturale, x, y
Rezultat:   cmmdc pentru cele doua numere naturale
*/
int cmmdc_extins(int a, int b, int& x, int& y)
{
    x = 0;
    y = 1;
    int u = 1;
    int v = 0;
    int q, r, m, n;
    while (a != 0)
    {
        q = b / a;
        r = b % a;
        m = x + u * q;
        n = y + v * q;
        b = a;
        a = r;
        x = u;
        y = v;
        u = m;
        v = n;
    }
    return b;
}

int main()
{
    int x = 0, y = 0, cmmdc = 0;
    cmmdc = cmmdc_extins(4, 6, x, y);
    std::cout << "cmmdc extins: " << cmmdc << ", x = " << x << ", y = " << y <<
std::endl;
    cmmdc = cmmdc_extins(10, 5, x, y);
    std::cout << "cmmdc : " << cmmdc << ", x = " << x << ", y = " << y << std::endl;
    cmmdc = cmmdc_extins(24, 63, x, y);
    std::cout << "cmmdc : " << cmmdc << ", x = " << x << ", y = " << y << std::endl;
    return 0;
}
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Rezolvare Pascal

```
{  
Descriere:   algoritmul lui Euclid in varianta extinsa  
Date:       a, b - numere naturale  
Rezultat:   cmmdc pentru cele doua numere naturale  
}
```

```
program cmmdc;  
function cmmdc_extins(a:integer; b:integer):integer;  
var x,y,u,v,q,r,m,n:integer;  
begin  
    x := 0;  
    y := 1;  
    u := 1;  
    v := 0;  
    while (a <> 0) do  
        begin  
            q := b div a;  
            r := b mod a;  
            m := x + u * q;  
            n := y + v * q;  
            b := a;  
            a := r;  
            x := u;  
            y := v;  
            u := m;  
            v := n;  
        end;  
    writeln('x = ',x,' y = ', y);  
    cmmdc_extins:=b;  
end;  
begin  
  
    writeln( 'cmmdc extins: ',cmmdc_extins(4, 6));  
    writeln( 'cmmdc extins: ',cmmdc_extins(10, 5));  
    writeln( 'cmmdc extins: ',cmmdc_extins(24, 63));  
end.
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Problema 10

O companie nou înființată de taxi are următorul model de taxare pentru șoferii săi: un șofer începe săptămâna cu 0 unități monetare, în timpul săptămânii fiecare șofer trebuie să predea companiei tot câștigul realizat în momentul în care are asupra sa  $N$  unități monetare, orice câștig suplimentar rămâne în posesia șoferului. Compania are angajați  $K$  șoferi, pentru fiecare se cunoaște numărul curselor efectuate și câștigul obținut. În nici un moment un șofer nu poate avea asupra sa mai mult de  $N-1$  unități monetare (decât imediat după o cursă) deoarece imediat ce are mai mult de  $N$  unități monetare trebuie să le predea companiei. Mai mult, dacă după ce predă  $N$  unități monetare, numărul său de unități monetare este din nou mai mare decât  $N$ , el va preda din nou  $N$  unități monetare până când va avea un câștig mai mic decât  $N$ . Determinați câștigul de la sfârșitul săptămânii pentru un șofer.

Să se citească de la tastatură valoarea  $N$ , numărul curselor pentru un șofer și câștigul pentru fiecare cursă. Programul va afișa câștigul obținut de șofer la sfârșitul săptămânii.

## Exemple

Date de intrare	Rezultat
50 4 1 2 3 4	10
50 5 10 20 30 40 50	0
50 3 874 9735 835	44

## Analiză

- aritmetică modulară (însușiri modulo)
- pe baza modului de efectuare a operațiilor în aritmetica modulară, următoarea relație se respectă:

$$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$$

## Rezolvare C++

```
/*  
Problema a fost compilată cu Microsoft Visual Studio varianta Community 2017  
*/  
#include <iostream>  
  
/*  
Descriere: calculează suma modulo  $n$  pentru două numere naturale  
Date:  $a, b, n$  - numere naturale,  
Rezultat:  $(a + b) \bmod n$   
*/  
int adunare_modulo(int a, int b, int n)  
{  
    return (a + b) % n;  
}
```

**Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică**  
**Consultații la Informatică pentru pregătirea concursului de admitere 2020**

---

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
/*
Descriere:   detrmna prin sume modulo castigul unui sofer
Date:       n - numar natural
Rezultat:   castigul unui sofer la sfarsitul saptamanii
*/
int determinaCastigSofer(int n)
{
    int curse;
    int castig = 0;
    int castig_cursa;
    std::cin >> curse;
    while (curse)
    {
        std::cin >> castig_cursa;
        castig = adunare_modulo(castig, castig_cursa, n);
        curse--;
    }
    return castig;
}

int main()
{
    int n = 1;

    while (n)
    {
        std::cin >> n;
        std::cout << "castigul soferului este: " << determinaCastigSofer(n) <<
std::endl;
    }

    return 0;
}
```



11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Rezolvare Pascal

```
program castig;
var n:integer;
{
Descriere:    calculeaza suma modulo n pentru doua numere naturale
Date:        a, b, n - numere naturale,
Rezultat:    (a + b) modulo n
}
function adunare_modulo(a:integer; b:integer; n:integer):integer;
begin
    adunare_modulo:= (a + b) mod n;
end;

{
Descriere:    detrmna prin sume modulo castigul unui sofer
Date:        n - numar natural
Rezultat:    castigul unui sofer la sfarsitul saptamanii
}
function determinaCastigSofer(n:integer):integer;
var curse, castig, castig_cursa:integer;
begin
    castig := 0;
    readln(curse);
    while (curse<>0) do
    begin
        readln(castig_cursa);
        castig := adunare_modulo(castig, castig_cursa, n);
        dec(curse);
    end;
    determinaCastigSofer:=castig;
end;
begin
    n := 1;

    while (n<>0) do
    begin
        readln(n);
        if (n<>0) then writeln('castigul soferului este: ', determinaCastigSofer(n));
    end;
end.
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

## Problema 11

Fie un număr  $n$ , se definește gradul numărului  $n$  ca fiind numărul de secvențe "101" din reprezentarea binară a acestuia. De exemplu, numărul 21 are gradul 2 (reprezentarea binară a lui 21 este 10101). Să se citească de la tastatură gradul  $k$  și mai multe numere (citirea se termină cu numărul 0), să se afișeze dacă un număr are gradul  $k$ .

## Exemple

Date de intrare	Rezultat
1 7 5 13 21 29 25 0	numere de grad cel puțin 1: 5 13 21 29
2 13 5 21 56 45 360 37 0	numere de grad cel puțin 2: 21 45 360

## Analiză

- operații la nivel de bit, izolarea celor mai puțin semnificativ trei biți din reprezentarea binară și verificarea dacă sunt o secvență 101, după care se deplasează la stânga cu una sau două poziții (în funcție de rezultatul comparației cu  $(101)_2 = (5)_{10}$ ).

## Rezolvare C++

```
/*  
Problema a fost compilata cu Microsoft Visual Studio varianta Community 2017  
*/  
#include <iostream>  
using namespace std;  
  
const int NRBITI = 32;  
int BINAR[NRBITI] = { 0 };  
  
/*  
Descriere: toate elementele vectorului BINAR = 0  
*/  
void resetBinar()  
{  
    for (int i = 0; i < NRBITI; ++i)  
    {  
        BINAR[i] = 0;  
    }  
}
```

**Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică**  
**Consultații la Informatică pentru pregătirea concursului de admitere 2020**

---

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
/*
 * Descriere:   reprezint un numar in baza doi
 * Date:       n - numarul natural de reprezentat in baza doi
 * Rezultat:   numarul de cifre din reprezentarea binara
 */
int calculReprezBinar(int n)
{
    resetBinar();
    int i;
    for (i = 0; n > 0; i++)
    {
        BINAR[i] = n % 2;
        n = n / 2;
    }
    return i + 1;
}

/*
Descriere:   reprezint un numar in baza doi, versiunea II folosind
operatii pe biti
Date:       n - numarul natural de reprezentat in baza doi
Rezultat:   numarul de cifre din reprezentarea binara
*/
int calculReprezBinar2(int n)
{
    resetBinar();
    int i = 0;
    /*
    se izoleaza fiecare cifra din reprezentarea binara a numarului
    si se stocheaza in vectorul BINAR pe pozitia ei
    */
    while (n)
    {
        BINAR[i] = n & 1;
        n >>= 1;
        i++;
    }
    return i;
}
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
/*
Descriere:   calcul determina gradul unui numar (numarul de secvente 101
             din reprezentarea binara)
Date:       n - numarul natural pentru care trebuie determinat gradul
Rezultat:   gradul numarului
*/
int determinaGrad(int n)
{
    int grad = 0;
    int nrCifre = 0;
    nrCifre = calculReprezBinar2(n);
    for (int i = nrCifre; i >= 2; i--)
    {
        if (BINAR[i] == 1 && BINAR[i - 1] == 0 && BINAR[i - 2] == 1)
            grad++;
    }
    return grad;
}

int main()
{
    int k, numar = 1;
    // se citeste gradul cautat
    cin >> k;
    /* sirul s va stoca numerele de grad cel puțin k */
    int s[100] = { 0 };
    int i = 0;
    /* citirea numerelor se termina cu citirea lui 0 */
    while (numar)
    {
        cin >> numar;
        if (k <= determinaGrad(numar))
        {
            s[i] = numar;
            i++;
        }
    }
    cout << "numere de grad cel puțin " << k << " :" << endl;
    for (int y = 0; y < i; ++y)
    {
        cout << s[y] << " ";
    }
    cout << endl;
    return 0;
}
```

O abordare mai eficientă poate fi imaginată prin izolarea celor mai puțin semnificativ trei biți din reprezentarea binară și verificarea dacă sunt o secvență 101, după care se deplasează la stânga cu una sau două poziții (în funcție de rezultatul comparației cu  $(101)_2 = (5)_{10}$ ).

**Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică**  
**Consultații la Informatică pentru pregătirea concursului de admitere 2020**

---

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
/*
Problema a fost compilata cu Microsoft Visual Studio varianta Community 2017
*/
#include <iostream>
using namespace std;

/*
Descriere:   calcul determina gradul unui numar (numarul de secvente 101
             din reprezentarea binara)
Date:       n - numarul natural pentru care trebuie determinat gradul
Rezultat:   gradul numarului
*/
int determinaGrad2(int n)
{
    int grad = 0;
    while (n)
    {
        /*
        se izoleaza cei mai putin semnificativ 3 biti si se verifica
        daca sunt o secventa 101
        */
        if ((n & 7) == 5)
        {
            grad++;
            n >>= 2;
        }
        else
        {
            n >>= 1;
        }
    }
    return grad;
}

int main()
{
    int k, numar = 1;
    // se citeste gradul cautat
    cin >> k;
    /* citirea numerelor se termina cu citirea lui 0 */
    while (numar)
    {
        cin >> numar;
        if (k == determinaGrad2(numar))
        {
            cout << "numarul " << numar << " are gradul " << k << endl;
        }
    }
    return 0;
}
```

Exemple:

Date de intrare si rezultat
-----------------------------

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

2 21 numarul 21 are gradul 2 7 5 29 45 numarul 45 are gradul 2 360 numarul 360 are gradul 2 0
3 362 numarul 362 are gradul 3 360 85 numarul 85 are gradul 3 0

## Rezolvare Pascal

```
program HelloWorld;

const NRBITI = 32;
var  BINAR:array [1..32] of integer;
var  k, numar,i,y:integer;
var  s:array [1..100] of integer;

procedure resetBinar();
var i:integer;
begin
for i:=1 to NRBITI do
begin
BINAR[i] := 0;
end;
end;

function calculReprezBinar(n: integer ):integer;
var i:integer;
begin
resetBinar();
i := 1;
while (n>0) do
begin
BINAR[i] := n mod 2;
n := n div 2;
end;
calculReprezBinar:=i+1;
end;

function calculReprezBinar2(n: integer):integer;
```

**Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică**  
**Consultații la Informatică pentru pregătirea concursului de admitere 2020**

---

11 ianuarie 2020

Lect. Dr. Mihai Suciu  
Asist. Dr. Adriana Coroiu

```
var i:integer;
begin
  resetBinar();
  i:= 1;
  while (n>0) do
  begin
    BINAR[i]:=n and 1;
    n:=n>>1;
    inc(i);
  end;
  calculReprezBinar2:=i;
end;

function determinaGrad2(n: integer ):integer;
var grad:integer;
begin
  grad:= 0;
  while (n>0) do
  begin
    if ((n and 7)=5) then
    begin
      inc(grad);
      n:= n>>2;
    end
    else
      n:= n>>1;
    end;
  determinaGrad2:=grad;
end;

function determinaGrad(n: integer ):integer;
var grad, nrCifre,i:integer;
begin
  grad:= 0;
  nrCifre:= 0;
  nrCifre := calculReprezBinar2(n);
  for i := nrCifre downto 3 do
    if ((BINAR[i]=1) and (BINAR[i-1]=0) and (BINAR[i-2]=1)) then
      inc(grad);

  determinaGrad:= grad;
end;

begin
  numar := 1;
  // se citeste gradul cautat
  readln(k);
  // varainta 1 fara siruri
  // se citesc elementele sirului, citirea se termina la introducerea
  numarului 0
  while (numar>0) do
  begin
    readln(numar);
    if (k = determinaGrad(numar)) then
    begin
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
        writeln('numarul ',numar,' are gradul= ',k);
    end;
end;
// varianta 2 cu siruri
i := 1;
numar:=1;
while (numar>0) do
begin
    readln(numar);
    if (k <= determinaGrad2(numar)) then
    begin
        s[i] := numar;
        inc(i);
    end;
end;
writeln( 'numere de grad ', k, ' :');
for y := 1 to i-1 do
    write(s[y], ' ');
end.
```

## Problema 12

Un număr  $n$  este "zâmbăreț" dacă duce la 1 după o secvență de pași unde în fiecare pas numărul este înlocuit cu suma pătratelor cifrelor ce formează numărul. Sa se scrie un program care citește mai multe numere până la citirea numărului 0 și determină câte numere "zâmbărețe" s-au citit.

### Exemple

numarul 19 este "zâmbăreț"

pas\_1:  $1+9^2=82$

pas\_2:  $64+4=68$

pas\_3:  $36+64=100$

pas\_4:  $1+0+0=1$

### Rezolvare C++

```
/*
Problema a fost compilata cu Microsoft Visual Studio varianta Community 2017
*/
#include <iostream>
using namespace std;
```



**Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică**  
**Consultații la Informatică pentru pregătirea concursului de admitere 2020**

---

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
/*
Descriere:   calcul suma patrate cifre numar
Date:       n - numar natural
Rezultat:   suma patrate numere
*/
int sumPatrate(int n)
{
    int suma = 0;
    while (n)
    {
        suma += (n % 10) * (n % 10);
        n /= 10;
    }
    return suma;
}

/*
Descriere:   determin daca un numar este zambaret
Date:       n - numar natural
Rezultat:   True - numarul e zambaret
            False - numarul nu este zambaret
*/
bool eNumarZambaret(int n)
{
    int nr1, nr2;
    // initializare numere cu n
    nr1 = nr2 = n;
    /*
    un numar nu este zambaret daca pe parcursul iteratiilor
    atinge aceeasi valoare succesiv
    */
    do
    {
        nr1 = sumPatrate(nr1);
        nr2 = sumPatrate(sumPatrate(nr2));
    } while (nr1 != nr2);

    // daca ambele numere sunt 1, return true
    return (nr1 == 1);
}
```

11 ianuarie 2020

Lect. Dr. Mihai Suciuc  
Asist. Dr. Adriana Coroiu

```
int main()
{
    int contor = 0;
    int numar = 1;
    while (numar)
    {
        cin >> numar;
        if (eNumarZambaret(numar))
        {
            cout << "numarul " << numar << " este zambaret" << endl;
            contor++;
        }
    }
    cout << "Am citit " << contor << " numere zambarete" << endl;
    return 0;
}
```

## Rezolvare Pascal

```
function sumPatrate(n:integer):integer;
var suma:integer;
begin
    suma := 0;
    while (n>0) do
    begin
        suma := suma+ (n mod 10) * (n mod 10);
        n :=n div 10;
    end;
    sumPatrate :=suma;
end;
function eNumarZambaret(n:integer):boolean;
var nr1, nr2:integer;
begin
    // initializare numere cun
    nr1 := n;
    nr2 := n;

    // un numar nu este zambaret daca pe parcursul iteratiilor
    // atinge aceeasi valoare succesiv

    repeat
        nr1 := sumPatrate(nr1);
        nr2 := sumPatrate(sumPatrate(nr2));
    until (nr1 = nr2);

    // daca ambele numere sunt 1, return true
    eNumarZambaret:= (nr1 = 1);
end;
var contor, numar:integer;
begin
    contor := 0;
    numar := 1;
```

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
while (numar>0) do
begin
  readln(numar);
  if (eNumarZambaret(numar)) then
  begin
    writeln(numar, ' este zambaret');
    inc(contor);
  end;
end;
writeln( 'Am citit ',contor,' numere zambarete');
end.
```

Exemple:

Date de intrare	Rezultat
49	23 este zambaret
19	49 este zambaret
100	19 este zambaret
11	100 este zambaret
0	Am citit 4 numere zambarete

## Probleme tip grilă

1.

a) Variabila  $x$  este de tip real. Care dintre următoarele expresii are valoarea 1 dacă și numai dacă numărul real memorat în variabila  $x$  aparține intervalului  $(5,8]$ ?

- a.  $(x < 8) \ \&\& \ (x \geq 5)$
- b.  $(x \leq 8) \ || \ (x > 5)$
- c.  $(x > 8) \ || \ (x \leq 5)$
- d.  $(x \leq 8) \ \&\& \ (x > 5)$

b) Variabilele  $x$  și  $y$  sunt tip int. Care dintre expresiile de mai jos are valoarea 1 dacă și numai dacă valorile întregi nenule memorate în variabilele  $x$  și  $y$  sunt egale?

- a.  $(x \% y == 0) \ \&\& \ (y \% x == 0) \ \&\& \ (x * y > 0)$
- b.  $(x \leq y) \ \&\& \ (y < x)$
- c.  $(x \leq y) \ || \ (y \leq x)$
- d.  $x * x == y * y$

2. Se da următoarea secvență de cod scrisă în pseudocod.

a)

**citește**  $n$  (număr natural)

Universitatea Babeş-Bolyai, Facultatea de Matematică și Informatică  
Consultații la Informatică pentru pregătirea concursului de admitere 2020

---

11 ianuarie 2020

Lect. Dr. Mihai Suci  
Asist. Dr. Adriana Coroiu

```
z ← 0
p ← 1
┌cât timp n>0 execută
| c ← n%10
| n ← [n/10]
| ┌dacă c%3=0 atunci
| | z ← z+p*(9-c)
| | p ← p*10
| └─┘
└─┘
scrie z
```

Care este rezultatul dacă se execută pentru i)  $n = 103?$  ii)  $n = 103456?$

a. 86, 863                      b. 96, 963                      c. 76, 763                      d. 96, 936

b)

citește  $x$  (număr natural nenul)

```
┌cât timp x>0 execută
| citește y (număr natural)
| ┌dacă x>y atunci
| | scrie x%10
| | altfel
| | scrie y%10
| └─┘
└─┘
x ← y
```

Care se va afișa după execuția codului dacă se citesc valorile 17 22 13 101 2 7 5 0?

a. 2211775                      b. 7231275                      c. 1211000                      d. 5771122

*Raspunsuri grile:*

*1a - d; 1b - a; 2a - b; 2b - a*