

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

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

Problema 1

Enunț

- Să se realizeze câte o funcție recursivă cu un singur parametru (numărul n) pentru:
- determinarea cifrei minime a lui n , se va returna cifra minimă;
 - determinarea cifrei pare maxime a lui n , se va returna cifra pară maximă sau -1 , dacă n nu are cifre pare;

Analiză

Vom scrie formulele recursive pentru cele două cerințe.

$$a) \text{cifraMinima}(n) = \begin{cases} n, & \text{dacă } n \text{ este format dintr-o singură cifră} \\ \min\{n \bmod 10, \text{cifraMinima}(\lfloor \frac{n}{10} \rfloor)\}, & \text{altfel} \end{cases}$$

O altă variantă recursivă pentru determinarea cifrei minime este:

$$\text{cifraMinima}(\overline{a_1 a_2 \dots a_{n-1} a_n}) = \begin{cases} n, & \text{dacă } n \text{ este format dintr-o singură cifră} \\ \text{cifraMinima}(\overline{a_1 a_2 \dots \min\{a_{n-1}, a_n\}}), & \text{altfel} \end{cases}$$

În această variantă apelul recursiv se face pentru numărul obținut prin înlocuirea ultimelor două cifre ale sale cu minimumul dintre acestea.

b)

$$\text{cifraParaMaxima}(n) = \begin{cases} n, & \text{dacă } n \text{ este format dintr-o singură cifră pară} \\ -1, & \text{dacă } n \text{ este format dintr-o singură cifră impară} \\ \text{cifraParaMaxima}(\lfloor \frac{n}{10} \rfloor), & \text{dacă } n \bmod 10 \text{ este impară} \\ \max\{n \bmod 10, \text{cifraParaMaxima}(\lfloor \frac{n}{10} \rfloor)\}, & \text{dacă } n \bmod 10 \text{ este pară} \end{cases}$$

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Specificarea funcțiilor

Funcția **cifraMinima(n)**:

Descriere: Returneaza cifra minima a unui numar dat.

Date: n – numar natural.

Rezultate: cifra minima a numarului dat.

Funcția **cifraParaMaxima(n)**:

Descriere: Returneaza cifra para maxima a unui numar dat.

Date: n – numar natural.

Rezultate: cifra para maxima a numarului dat sau -1 daca numarul are doar cifre impare.

Implementare

Varianta C++

```
#include <iostream>

using namespace std;

/*
Descriere: Returneaza cifra minima a unui numar dat.
Date: n – numar natural.
Rezultate: cifra minima a numarului dat.
*/
int cifraMinima_V1(int n)
{
    if (n <= 9) // cazul in care numarul este format dintr-o singura cifra
        return n;
    int cifraMinima_restul_numarului = cifraMinima_V1(n / 10);
    return (n % 10 < cifraMinima_restul_numarului ? n % 10 :
cifraMinima_restul_numarului);
}

/*
Descriere: Returneaza cifra minima a unui numar dat.
Date: n – numar natural.
Rezultate: cifra minima a numarului dat.
*/
int cifraMinima_V2(int n)
{
    if (n <= 9) // cazul in care numarul este format dintr-o singura cifra
        return n;
    int minim_ultimele_doua_cifre = n % 10;
    int penultima_cifra = (n / 10) % 10;
    if (penultima_cifra < minim_ultimele_doua_cifre)
        minim_ultimele_doua_cifre = penultima_cifra;
    return cifraMinima_V2((n/100) * 10 + minim_ultimele_doua_cifre);
}
```

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
/*
Descriere: Returneaza cifra para maxima a unui numar dat.
Date : n - numar natural.
Rezultate : cifra para maxima a numarului dat.
*/
int cifraParaMaxima(int n)
{
    if (n <= 9)          // daca n este format dintr-o singura cifra
    {
        if (n % 2 == 0)      // daca este par
            return n;
        else
            return -1;
    }

    int ultima_cifra = n % 10;
    if (ultima_cifra % 2 != 0)
        return cifraParaMaxima(n / 10);
    int cifraParaMaxima_restul_numarului = cifraParaMaxima(n / 10);
    return (ultima_cifra > cifraParaMaxima_restul_numarului ? ultima_cifra :
cifraParaMaxima_restul_numarului);
}

int main()
{
    int n = 0;
    cout << "Introduceti numarul: ";
    cin >> n;

    cout << "Cifra minima a numarului " << n << " este: " << cifraMinima_V1(n) <<
endl;
    cout << "Cifra minima a numarului " << n << " este: " << cifraMinima_V2(n) <<
endl;
    cout << "Cifra para maxima a numarului " << n << " este: " << cifraParaMaxima(n)
<< endl;

    return 0;
}
```

Varianta Pascal

```
{
Descriere: Returneaza cifra minima a unui numar dat.
Date: n - numar natural.
Rezultate: cifra minima a numarului dat.
}
function cifraMinima_V1(n: longint): integer;
var cifraMinima_restul_numarului: integer;
begin
```

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
    if (n <= 9) then // cazul in care numarul este format dintr-o singura
cifra
        cifraMinima_V1 := n
    else
    begin
        cifraMinima_restul_numarului := cifraMinima_V1(n div 10);
        if ((n mod 10) < cifraMinima_restul_numarului) then
            cifraMinima_V1 := n mod 10
        else
            cifraMinima_V1 := cifraMinima_restul_numarului;
    end;
end;

function cifraMinima_V2(n: longint): integer;
var ultima_cifra, penultima_cifra: integer;
begin
    if(n > 9) then
    begin
        ultima_cifra := n mod 10;
        penultima_cifra := (n div 10) mod 10;
        if (penultima_cifra < ultima_cifra)
            then cifraMinima_V2 := cifraMinima_V2(n div 10)
            else cifraMinima_V2 := cifraMinima_V2(((n div 100) * 10) +
ultima_cifra);
        end
    else cifraMinima_V2 := n;
end;

{
Descriere: Returneaza cifra para maxima a unui numar dat.
Date : n - numar natural.
Rezultate : cifra para maxima a numarului dat.
}
function cifraParaMaxima(n: longint): integer;
var ultima_cifra, cifraParaMaxima_restul_numarului: integer;
begin
    if (n <= 9) then {daca n este format dintr-o singura cifra}
    begin
        if (n mod 2 = 0) then {daca este par}
            cifraParaMaxima := n
        else
            cifraParaMaxima := -1;
        end
    else
    begin
        ultima_cifra := n mod 10;
        if (ultima_cifra mod 2 <> 0) then
            cifraParaMaxima := cifraParaMaxima(n div 10)
        else
        begin
            cifraParaMaxima_restul_numarului := cifraParaMaxima(n div 10);
            if (ultima_cifra > cifraParaMaxima_restul_numarului) then
```

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
        cifraParaMaxima := ultima_cifra
    else
        cifraParaMaxima := cifraParaMaxima_restul_numarului;
    end;
end;
end;

var n: longint;
begin

    write('Introduceti numarul: ');
    readln(n);
    writeln('Cifra minima este: ', cifraMinima_V1(n));
    writeln('Cifra minima este: ', cifraMinima_V2(n));
    writeln('Cifra para maxima este: ', cifraParaMaxima(n));

end.
```

Problema 2

Enunț

Scrieti un subprogram recursiv care determina numarul de aparitii ale unei cifre in reprezentarea zecimala a numarului natural n.

Analiză

Formula recursiva pentru aceasta problema este:

$$aparitii(n, c) = \begin{cases} 1, & \text{daca } n < 10 \text{ si } n = c \\ 0, & \text{daca } n < 10 \text{ si } n \neq c \\ 1 + aparitii\left(\frac{n}{10}, c\right), & \text{daca } n > c \text{ si } n \bmod 10 = c \\ aparitii\left(\frac{n}{10}, c\right), & \text{daca } n > c \text{ si } n \bmod 10 \neq c \end{cases}$$

Specificarea subalgoritmului

Funcția **aparitii(n,c)**:

Descriere: Returneaza numarul aparitiilor cifrei c in numarul n

Date: n – numar natural, $0 \leq c \leq 9$, c- numar natural.

Rezultate: numarul de aparitii ale cifrei c in numarul n.

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Implementare

Varianta C++

```
#include <iostream>
using namespace std;
/*
Descriere: Returneaza numarul de aparitii ale cifrei c in numarul n.
Date : n - numar natural, c umar natural 0<=c<=9.
Rezultate : numarul de aparitii ale cifrei c in numarul n.
*/

int aparitii(int n, int c)
{
    // daca n are o singura cifra returnez 1 daca cifra = c, 0 altfel
    if (n < 10) return n == c;
    //daca ultima cifra e c o adauga la numaratoare
    else if (n % 10 == c) return 1 + aparitii(n / 10, c);
    else return aparitii(n / 10, c);
}

int main() {
    int n, c;
    cout << "Introduceti numarul: ";
    cin >> n;
    cout << "Introduceti cifra: ";
    cin >> c;
    cout << aparitii(n, c);
    return 0;
}
```

Varianta Pascal

```
program aparitii;
{Descriere:
  functia determina numarul de aparitii ale cifrei c in numarul n
}
function aparitii(n:integer; c:integer):integer;
begin
  if (n<10) then
    if n=c then aparitii:=1 {daca numarul are o singura cifra
                           {si cifra e egala cu c, marcam o aparitie}
    else aparitii:=0 {altfel, incepem numararea aparitiilor de la 0}
  else
    {daca ultima cifra a numarului e chiar c, incrementam numarul aparitiilor}
    if n mod 10=c then aparitii:=1+aparitii(n div 10, c)
```

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
{daca ultima cifra a lui n e diferita de c, continui procesul recursiv}
  else aparitii:=aparitii(n div 10,c);
end;

var n,c:integer;
{program principal}
begin
write('Introduceti numarul n:');
  readln(n);
write('Introduceti cifra c:');
  readln(c);
  writeln(aparitii(n,c));
end.
```

Problema 3

Enunț

Scrieti o functie recursiva care primeste ca parametru un numar natural n si returneaza numarul obtinut din n prin eliminarea cifrelor pare.

Exemplu: Daca $n=4536597$, atunci $rez = 53597$.

Analiză

Formula recursiva:

$$\text{elimCifrePare}(n) = \begin{cases} 0, & \text{daca } n = 0 \\ \text{cifrePare}\left(\left\lfloor \frac{n}{10} \right\rfloor\right), & \text{daca } n \text{ numar par} \\ \text{cifre}\left(\left\lfloor \frac{n}{10} \right\rfloor\right) * 10 + n \bmod 10, & \text{daca } n \text{ numar impar} \end{cases}$$

Specificarea subalgoritmului

Funcția **elimCifrePare(n)**:

Descriere: Returneaza numarul obtinut prin eliminarea cifrelor pare din numarul n

Date: n – numar natural

Rezultate: numarul obtinut prin eliminarea cifrelor pare din numarul n

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Implementare

Varianta C++

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

/*Determina numarul obtinut prin eliminarea cifrelor pare din numarul natural n
  Date: n numar natural
  Rezultate: numarul obtinut din n dupa eliminarea cifrelor pare
*/
int elimCifrePare(int n)
{
    if (n == 0) return 0;
    else
        /*daca ultima cifra e para, o ignor si continui procesul*/
        if (n % 2 == 0) return elimCifrePare(n / 10);
        /*daca ultima cifra e impara, o folosesc la contruirea numarului format doar
          din cifre impare*/
        else return elimCifrePare(n / 10) * 10 + n % 10;
}

int main() {
    int n;
    cout << "Dati numarul n: ";
    cin >> n;
    cout << elimCifrePare(n);
    return 0;
}
```

Varianta Pascal

```
Program eliminare;

{Functia determina numarul obtinut prin eliminarea cifrelor pare din numarul
natural n
Date: n numar natural
Rezultate: numarul obtinut din n dupa eliminarea cifrelor pare
}

function elimCifrePare(n:integer):integer;
begin
    if n=0 then elimCifrePare:=0
    else
        {daca ultima cifra e para, o ignor si continui procesul}
        if n mod 2=0 then elimCifrePare:=elimCifrePare( n div 10)
        {daca ultima cifra e impara, o folosesc la contruirea numarului format doar
          din cifre impare}
```


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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
        else elimCifrePare:=elimCifrePare(n div 10)*10+n mod 10;
    end;

var n:integer;
begin
    write('Dati numarul n: ');
    readln(n);
    writeln('Numarul obtinut prin eliminarea cifrelor pare este ',
elimCifrePare(n));
end.
```

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Problema 4

Enunț

Fie i, j, k trei numere naturale, $j > 0, k > 1$. Să scrie un subprogram care returnează restul împărțirii numărului (i^j) la k , deci $(i^j) \bmod k$ (iterativ și recursiv).

Exemple:

$$(100^{100}) \bmod 7 = 2$$

$$(125^{199}) \bmod 999 = 800$$

$$(2^{10}) \bmod 9 = 7$$

Analiză

- practic se înmulțesc resturile modulo k ; nu e nevoie de calculul expresiei i^j ;
- se calculează la fiecare pas de înmulțire restul produsului (modulo k);

Aceasta deoarece se știe că:

$$(a * b) \bmod c = (a \bmod c * b \bmod c) \bmod c$$

Atunci:

$$a^x \bmod c = (a \bmod c * a^{x-1} \bmod c) \bmod c$$

Pentru varianta recursivă, formula recursivă se deduce din formula anterioară astfel:

$$rest(i, j, k) = \begin{cases} 1, & \text{daca } j = 0 \\ ((i \bmod k) * rest(i, j - 1, k)) \bmod k & \end{cases}$$

Specificarea funcției

Funcția **Rest(i, j, k)**:

Descriere: returnează restul împărțirii lui (i^j) la k .

Date: i, j, k - numere naturale

Rezultate: R un număr natural: $R \in \{0, 1, \dots, k-1\}$

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Implementare

Varianta iterativă C++

```
/*
Descriere: returneaza restul impartirii lui (i^j) la k.
Date: i,j,k - numere naturale, j>0, k>1
Rezultate: R un numar natural: R in {0,1,...k-1}.
*/
int Rest(int i, int j, int k)
{
    int r = i % k;
    int rest = 1;
    while (j > 0)
    {
        rest = (rest * r) % k;
        j--;
    }
    return rest;
}
```

Varianta recursivă C++

```
/*
Descriere: returneaza restul impartirii lui (i^j) la k.
Date: i,j,k - numere naturale
Rezultate: R un numar natural: R in {0,1,...k-1}.
*/
int Rest(int i, int j, int k)
{
    if (j == 0)
        return 1;
    return ((i % k) * Rest(i, j - 1, k)) % k;
}
```

Varianta iterativă Pascal

```
function Rest(i,j,k:integer):integer;           {functie ce determina restul}
var R,iModK:integer;
begin                                           {e suficient sa inmultim resturile}
    iModK:=i mod k;                             {expresia i Mod k e constanta in ciclu}
    R :=1;
    While(j>0) do
        begin
            R:=(R*(iModK)) mod k;               {atat lui i cat si produsului se aplica mod}
            j:=j-1;
        end;
    Rest:=R;
```

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

end;

Varianta recursivă Pascal

```
function Rest (i,j,k:integer):integer);  
begin  
    if(j>0) then Rest:= ((i mod k)*Rest(i,j-1,k)) % k  
    else Rest:=1;  
end;
```

Exemple

Date de intrare	Rezultate
100, 100, 7	2
125, 199, 999	800
2, 10, 9	7
8, 0, 100	1
5, 151, 5	0

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Problema 5

Enunț

Fie x, y două numere naturale. Să scrie un subprogram care determină dacă două numere date sunt **asemenea**, fără a folosi tablouri. (Două numere naturale sunt **asemenea** dacă au aceleași cifre: 2131 e **asemenea** cu 32211 pentru că mulțimea cifrelor este aceeași: $\{1, 2, 3\}$).

Analiză

Problema are o rezolvare simplă dacă se folosește conceptul de vector de apariție pentru cifrele unui număr:

- se determină vectorii de apariție pentru x și y ;
- se compară apoi vectorii de apariție și se decide asemănarea.

Fără utilizarea vectorilor ar trebui să vedem dacă fiecare cifră a lui x este în y și invers.

Vom face 2 subprograme:

- Un subprogram care verifică dacă fiecare cifră a unui număr se află printre cifrele altui număr.
- Un subprogram care apelează apoi de 2 ori primul subprogram.

Specificarea subalgoritmilor

Functia **CifreAinB** (A, B):

Descriere: verifică dacă fiecare cifră a lui A apare în reprezentarea zecimală a lui B .

Date: $A, B > 0$ numere naturale

Rezultate: true, dacă mulțimea cifrelor lui A este inclusă sau egală cu mulțimea cifrelor lui B ;
false, în caz contrar

Functia **Asemenea** (X, Y):

Descriere: verifică dacă X și Y sunt asemenea.

Date: $X, Y > 0$ numere naturale

Rezultate: true, dacă X și Y sunt asemenea
false, în caz contrar

Implementare

Varianta C++

```
/*
```

```
Descriere: verifica daca fiecare cifra a lui A este in multimea cifrelor lui B.
```

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
Date: A, B > 0 numere naturale
Rezultate: true, daca multimea cifrelor lui A este inclusa in multimea cifrelor lui B;
           false, in caz contrar
*/
bool CifreAinB(long a, long b)
{
    int copieB = b; // se retine o copie a lui B
    while (a > 0) // vom verifica daca fiecare cifra a lui A este in B
    {
        int ultima_cifra_A = a % 10;
        while (copieB > 0 && ultima_cifra_A != copieB % 10)
            copieB /= 10;
        if (copieB == 0) // daca s-a ajuns la 0, cifra curenta a lui A nu este
            in B
                return false;
        copieB = b; // se reinitializeaza copia cu numarul initial B
        a /= 10; // se trece la urmatoarea cifra a lui A
    }
    return true;
}

/*
Descriere: verifica dubla incluziune a cifrelor lui x si y.
Date: x, y > 0 numere naturale
Rezultate: true, daca CifreAinB(x, y) este true SI CifreAinB(y,x) este true;
           false, in caz contrar
*/
bool Asemenea(long x, long y)
{
    return (CifreAinB(x, y) && CifreAinB(y, x));
}
```

Varianta Pascal

```
function CifreAinB(A,B:longint):boolean;
var CopieB :longint;
    UcifA :byte;
begin
    CopieB :=B; //se retine o clona a lui B in CopieB
    CifreAinB:=true; //presupunem ca incluziunea exista
    while (A>0) do
        begin
            UcifA:=A mod 10; //verifica fiecare cifra a lui A daca e in B
            B :=CopieB; //la fiecare cifra a lui A se reface B
            while (B>0) and (UcifA<>(B mod 10)) do B:=B div 10;
            if (B=0) //daca B=0 => cifra curenta a lui A nu e in B
                then
                    begin
                        CifreAinB:=false; //se returneaza false
                        A :=0;
                    end
                else A:=A div 10; //daca B>0, cifra curenta a lui A e in B
            end
        end
end
```

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
end;                                     {se trece la următoarea cifra a lui A}  
end;
```

```
function Asemenea(X,Y:longint):boolean;  
begin  
  if (CifreAinB(X,Y)) and (CifreAinB(Y,X))  
    then Asemenea:=true  
    else Asemenea:=false;  
end;
```

Exemple

Date de intrare	Rezultate
1222331, 123	true
122235, 123	false
5656565, 56	true
5656565, 5	false
1, 8	false

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Problema 6

Enunț

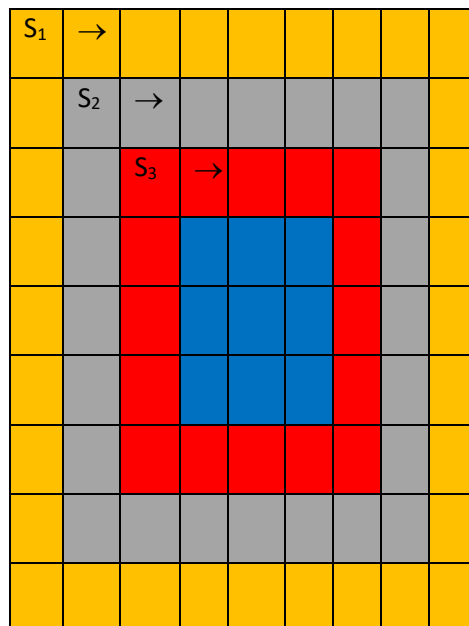
Un domeniu pătrat (cu albastru), ca cel din figură trebuie apărat. Domeniul e împărțit în pătrățele. Latura domeniului este de n pătrățele, $n \geq 7$. Pe culorile colorate patrulează 3 soldați, care încep patrularea din colțul stânga sus în același moment (poziția inițială). În exemplul de mai jos avem $n=9$.

Cei 3 soldați (garda) se vor schimba cu alți 3 soldați în momentul în care ajung în poziția inițială, simultan.

Soldații patrulează în sensul acelor de ceasornic fiecare cu aceeași viteză constantă, și fiecare pe culoarul său.

Să se implementeze un program care determină numărul de circuite complete pentru fiecare soldat (un circuit complet este făcut de un soldat oarecare, pe același culoar până ajunge în poziția inițială), pentru schimbarea gărzii.

S_1 patrulează pe culoarul galben, S_2 pe gri, S_3 pe roșu. Albastrul este cetatea care trebuie apărată.



Analiză

- Considerăm doi soldați care patrulează circuite de lungimi diferite, măsurate în unități (o unitate este un pătrățel). Un circuit are lungime L_1 , iar al doilea lungime L_2 unități.

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

- Cei doi vor ajunge concomitent în punctul inițial după ce parcurg $X = \text{cmmmc}(L1, L2)$ unități. Astfel, primul va parcurge $X/L1$ circuite, iar al doilea $X/L2$.
- Aceeași idee se aplică pentru trei soldați.

Specificarea subalgoritmilor

Funcția **cmmmc(a, b)**:

Descriere: Calculează cel mai mic multiplu comun a două numere naturale nenule.

Date: a, b – numere naturale nenule.

Rezultate: cel mai mic multiplu comun al celor două numere.

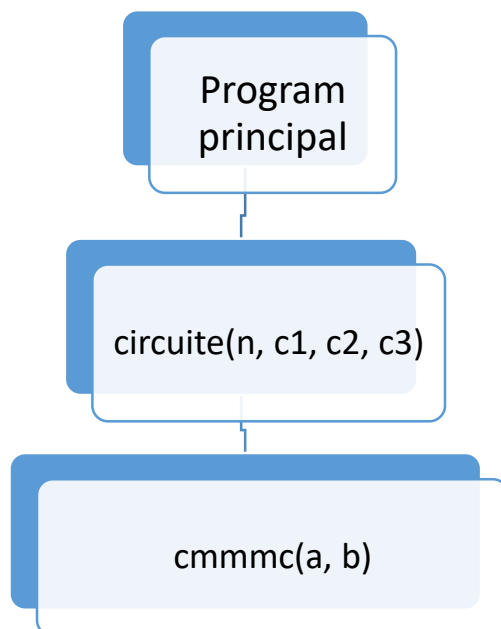
Subalgoritm **circuite(n, c1, c2, c3)**:

Descriere: Calculează numărul de circuite complete pentru fiecare soldat, până la întâlnire.

Date: n – latura domeniului.

Rezultate: c1, c2, c3 – numărul de circuite parcurse de către cei trei soldați.

Proiectare



9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

Implementare

Varianta C++

```
#include <iostream>

using namespace std;

/*
Descriere: Calculeaza cel mai mic multiplu comun a doua numere naturale nenule.
Date: a, b - numere naturale nenule.
Rezultate: cel mai mic multiplu comun al celor doua numere.
*/
int cmmmc(int a, int b)
{
    if (a < b)
    {
        int aux = a;
        a = b;
        b = aux;
    }

    // cel mai mare numar se aduna repetat cu el insusi
    // si se verifica divizibilitatea cu celalalt
    int rez = a;
    while (rez % b != 0)
        rez += a;

    return rez;
}

/*
Descriere: Calculeaza numarul de circuite complete pentru fiecare soldat, pana la
intalnire.
Date: n - latura domeniului.
Rezultate: c1, c2, c3 - numarul de circuite parcurse de catre cei trei soldati.
*/
void circuite(int n, int& c1, int& c2, int& c3)
{
    int l1 = 4 * (n - 1); // lungimea culoarului exterior
    int l2 = 4 * (n - 3); // lungimea culoarului din mijloc
    int l3 = 4 * (n - 5); // lungimea culoarului interior

    int intalnire = cmmmc(cmmmc(l1, l2), l3); // cmmmc al lui l1, l2, l3
    c1 = intalnire / l1;
    c2 = intalnire / l2;
    c3 = intalnire / l3;
}

int main()
{
    int n = 0;
    cout << "Introduceti latura domeniului: ";
```

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
    cin >> n;

    int c1 = 0, c2 = 0, c3 = 0;
    circuite(n, c1, c2, c3);
    cout << "Soldatul S1 va face " << c1 << " circuite pana la intalnire." << endl;
    cout << "Soldatul S2 va face " << c2 << " circuite pana la intalnire." << endl;
    cout << "Soldatul S3 va face " << c3 << " circuite pana la intalnire." << endl;

    return 0;
}
```

Varianta Pascal

```
function cmmmc(a,b:integer): integer;
                                {se aduna cel mai mare repetat cu el insusi,}
                                {si se verifica divizibilitatea cu celalalt}
var aux, rez: integer;
begin
    if(a < b)then
        begin
            aux := a;
            a := b;
            b := aux;
        end;

    rez := a;
    while (rez Mod b) > 0 do
        rez := rez + a;
    cmmmc := rez;
end;

procedure circuite(n:integer; var c1, c2, c3:integer);
var l1,l2,l3,intalnire: integer;      {nr. de patratele parcurse pentru un
circuit complet de cei 3 soldati}
begin
    l1:=4*(n-1);
    l2:=4*(n-3);
    l3:=4*(n-5);
    intalnire:=cmmmc(cmmmc(l1,l2),l3);  {cel mai mic multiplu comun al lui
p1,p2,p3}

    c1:=intalnire div l1;                {nr de circuite pentru fiecare soldat}
    c2:=intalnire div l2;
    c3:=intalnire div l3;
end;

var n, c1, c2, c3: integer;
begin
```

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

```
write('Introduceti latura domeniului: ');  
readln(n);  
circuite(n, c1, c2, c3);  
writeln('Soldatul S1 va face ', c1, ' circuite pana la intalnire.');
```

end.

Observatie: Alternativ, se poate lua in considerare calculul cmmmc folosind cmmdc, astfel:

$$cmmmc(a,b) = \frac{a * b}{cmmdc(a,b)}$$

Probleme tip grilă

1. Se considera subalgoritmul F care primeste parametrul n, n numar natural.

<p>Subalgoritmul F (n) :</p> <p>Daca (n<10) atunci</p> <p> returneaza n;</p> <p>sfDaca</p> <p>u=n mod 10;</p> <p>r=F(n div 10);</p> <p>daca (u<r) atunci</p> <p> returneaza u;</p> <p>sfDaca</p> <p> returneaza r;</p> <p>SfSubalgoritm</p>	<p>Care este rezultatul executiei subalgoritmului F?</p> <p>a) Cea mai mare cifra a lui n</p> <p>b) Cea mai mică cifra a lui n</p> <p>c) Cea mai mare cifra pară a lui n</p> <p>d) Cea mai mică cifra pară a lui n</p>
---	---

2. Se considera subalgoritmii urmatoari. Alegeti subalgoritmii care interschimba valorile a și b.

<p>A.</p> <p>Subalgoritmul F (a,b) :</p> <p> a=a-b;</p> <p> b=b+a;</p> <p> a=b-a;</p>	<p>B.</p> <p>Subalgoritmul F (a,b) :</p> <p> a=a+b;</p>
--	--

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
 Lect. Dr. Diana Cristea

<p>SfSubalgoritm</p>	<p>b=a-b; a=a-b; SfSubalgoritm</p>
<p>C.</p> <p>Subalgoritmul F (a,b) : a=a div b; b=a*b; a=b div a; SfSubalgoritm</p>	<p>D.</p> <p>Subalgoritmul F (a,b) : a=a*b; b=a div b; a=a div b; SfSubalgoritm</p>
<p>E.</p> <p>Subalgoritmul F (a,b) : b=b-a; b=a*b; a=b; SfSubalgoritm</p>	

3. Se considera subalgoritmul **calcul (n)**, n numar natural.

<p>Subalgoritmul calcul(n):</p> <p>n←50; i←3; Cattimp i≤ n executa</p> <p> i←i+5; SfCatTimp</p> <p>Returneaza i; SfSubalgoritm</p>	<p>Care este valoarea returnata dupa executia subalgoritmului?</p> <p>a) 50 b) 53 c) 48 d) 90</p>
---	---

4. Se considera subalgoritmul F care are un parametru n, n numar natural.

<p>Subalgoritmul F(n) :</p> <p>daca (n<10) atunci</p> <p>daca n mod 2=0 atunci</p> <p> returneaza n; altfel returneaza -1;</p>	<p>Care este rezultatul executiei subalgoritmului?</p> <p>a) Returneaza cea mai mare cifra a lui n, sau -1 b) Returneaza cea mai mare cifra pară a lui n, sau -1</p>
---	--

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

<pre>sfDaca Daca (n mod 2=0) atunci returneaza n mod 10 altfel returneaza -1; sfDaca u=n mod 10; m=F(n div 10); Daca (u>m) atunci returneaza u; altfel returneaza m; SfSubalgoritm</pre>	<p>c) Returneaza cea mai mare cifră a lui n</p> <p>d) Returneaza cea mai mică cifră pară a lui n, sau -1</p>
---	--

5. Ce valori vor avea a, b și k la sfârșitul secvenței de instrucțiuni?

<pre>int a=20, b=70, k=0; if(a<b) { b*=a; a=b/a; b/=a; } while(a>=b){ a-=b; k+=2; }</pre>	<p>a) 70,70,5</p> <p>b) 70,20,2</p> <p>c) 70,20,6</p> <p>d) 10,30,4</p> <p>e) 10,20,6</p>
---	---

6. Care este efectul executiei subalgoritmului SP (n), n numar natural?

<pre>void SP(long n) { for (int i = 2; i <= n; i++) { int c = 0; while (!(n%i)) {</pre>	<p>a) Afișează perechi de numere prime ($n \geq 2$).</p> <p>b) Afișează numerele prime până la n ($n \geq 2$).</p>
--	--

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

<pre>n /= i; c += 1; } if (c) cout << i << " " << c << endl; } }</pre>	<p>c) Afișează factorii primi ai lui n ($n \geq 2$).</p> <p>d) Afișează factorii primi ai lui n și puterile lor ($n \geq 2$).</p> <p>e) Subprogramul nu afișează nimic, dacă $n < 2$.</p>
--	---

7. Care este efectul executiei urmatoarei secvente de instructiuni?

<pre>citește a,b {numere întregi} x ← 1 cât timp (a > 0) și (b > 0) execută dacă (a mod 10) < (b mod 10) atunci x ← 0 sfDacă a ← [a/10] b ← [b/10] sfCât timp dacă (x = 1) și (b = 0) atunci scrie "DA" altfel scrie "NU" sfDacă</pre>	<p>a) Afișează „DA” pentru perechi de numere strict pozitive.</p> <p>b) Afișează „NU” pentru perechi de numere strict negative.</p> <p>c) Afișează „DA” dacă $a < b$.</p> <p>d) Afișează „DA” dacă fiecare cifră a lui a este mai mare sau egala cu cifra corespondentă a lui b (unități, zeci, etc.) și numărul cifrelor lui b este mai mic sau egal cu numărul cifrelor lui a dacă $a, b > 0$.</p> <p>e) Afișează „DA” pentru $a=34, b=12$</p>
---	---

8. Care este efectul executiei urmatoarei secvente de instructiuni pentru șirul: 5, 24, 5, 25, 5, 26, 0?

<pre>citește x {x natural} nr ← 0 s ← 0 cât timp x ≠ 0 execută nr ← nr + 1 dacă nr mod 2 = 0 atunci s ← s + x mod 10 sfDacă citește x sfCât timp scrie s, nr</pre>	<p>a) Afișează 0 0.</p> <p>b) Afișează 15 6.</p> <p>c) Afișează 10 9.</p> <p>d) Afișează 10 6.</p> <p>e) Afișează suma ultimelor cifre ale numerelor de rang (indice) par din șir și numărul de numere nenule citite</p>
--	--

9. Se considera urmatoarea secventa de instructiuni.

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
 Lect. Dr. Diana Cristea

<p>citește $n, c \{n, \text{natural} > 0, c \text{ cifra}\}$ $z \leftarrow 0$ cât timp $n \bmod 10 = c$ execută $n \leftarrow \lfloor n/10 \rfloor$ $z \leftarrow z+1$ sfCât timp scrie z</p>	<p>a) Afișează 3 pentru $n=123$ și $c=3$; b) Afișează 2 pentru $n=12003$ și $c=0$; c) Afișează 1, pentru $n=c=1$; d) Afișează 4, pentru $n=1277771$ și $c=7$ e) Afișează 3, pentru $n=12555$ și $c=5$.</p>
---	---

10. Fie subalgoritmul $g(a,b)$, unde a și b sunt numere naturale nenule:

<p>Subalgoritmul $g(a,b)$: dacă $(a=b)$ atunci returnează a; SfDacă Dacă $(a>b)$ atunci $g(a-b,b)$; altfel $g(a,b-a)$; sfDacă SfSubalgitm</p>	<p>Precizați de câte ori se autoapeleaza subalgoritmul g în următoarea secvență de instrucțiuni:</p> <p>$a \leftarrow 4$ $b \leftarrow 3$ $z \leftarrow g(x, y)$</p> <p>a) de 4 ori b) de 3 ori c) de o infinitate de ori d) niciodată</p>
---	--

11. Fie variabilele booleene a , b și c cu valorile de adevăr TRUE, TRUE și FALSE. Evaluați expresia:

NOT a OR NOT b AND c

Raspuns: FALS (se tine cont de prioritatea operatorilor)

12. Se considera subalgoritmul calcul, cu parametrul natural n .

<p>Subalgoritmul Calcul (n): Dacă $(n=0)$ atunci returnează 0; SfDacă Dacă $n \bmod 2=1$ atunci returnează calcul ($n \text{ div } 10$); SfDacă</p>	<p>Pentru care dintre valorile parametrului n subalgoritmul va returna valoarea 246? a) 23457 b) 234567 c) 24769</p>
--	--

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
 Lect. Dr. Diana Cristea

returnează calcul($n \text{ div } 10$)*10 + $n \text{ mod } 10$; SfSubalgoritm	d) 294576
--	-----------

13. Se considera subalgoritmul prim(n,d), n și d - numere naturale nenule, care verifica daca numarul n este prim.

<p>Subalgoritmul prim(n, d): dacă ($n < 2$) atunci returnează 0; sfDacă dacă $d=1$ atunci returnează 1; sfDacă Dacă ($n \text{ mod } d=0$) atunci returnează 0; sfDacă returnează prim($n,d-1$); SfSubalgoritm</p>	<p>Cum ar trebui efectuat apelul pentru ca subalgoritmul să returneze răspunsul corect: a) Prim(n,n) b) Prim($n,1$) c) Prim($n,n/2$) d) Prim($n/2,2$)</p>
--	---

14. Se considera următoarele expresii logice:

<p>1) ($\text{true} \ \&\& \ \text{true}$) false 2) ($\text{false} \ \&\& \ \text{true}$) true 3) ($\text{false} \ \&\& \ \text{true}$) false true 4) ($5 > 6$ $4 > 3$) && ($7 > 8$) 5) $!(7 > 6$ $3 > 4$)</p>	<p>Care dintre expresii au valoare de adevar TRUE? a) 1, 5 b) 1,2,5 c) 2,4 d) 1,2,3</p>
---	---

15. Se considera subalgoritmul ghici(n), unde n este un numar natural.

<p>Subalgoritmul ghici (n): $f \leftarrow 0, k \leftarrow 0$; pentru $c \leftarrow 0;9$; executa $x \leftarrow n; k \leftarrow 0$; cattimp ($x > 0$) executa daca ($x \text{ mod } 10 = c$) atunci $k \leftarrow k+1$;</p>	<p>Sa se determine rezultatul returnat de executarea subalgoritmului: a) Numărul de cifre al numărului n b) Frecvența fiecărei cifre din reprezentarea zecimală a numărului n</p>
---	---

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

9 noiembrie 2019

Lect. Dr. Adriana Guran
Lect. Dr. Diana Cristea

<pre>sfDaca x ← x div 10; sfCat daca (k>f) atunci f ← k; sfDaca sfPentru returneaza f; SfSubalgoritm</pre>	<p>c) Numarul de aparitii ale cifrei cu frecventa maxima in reprezentarea zecimala a numarului n</p> <p>d) Cifra maxima din reprezentarea zecimala a numarului n</p>
--	--