Consultații pentru elevii de liceu
organizate de Facultatea de Matematică şi Informatică
pentru pregătirea concursului Mate-Info și concursului de admitere 2021

14.11.2020, 13.00 – 14.15

Lect. Dr. Adriana COROIU

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

Problema

Enunț:

Sa se 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

$$\text{functie1}(n) = \begin{cases} n, & n \leq 1 \\ \text{functie1}(n-1) + \text{functie1}(n-2), & n > 1 \end{cases}$$

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

```cpp
#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 functie1(int n)
{
    if (n <= 1)
        return n;
    return functie1(n-1) + functie1(n-2);
}


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

    std::cout <<"Rezultat= "<<functie1(n)<<std::endl;
    return 0;

}
```

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

```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;
}

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 problema1iterativ;
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.
```

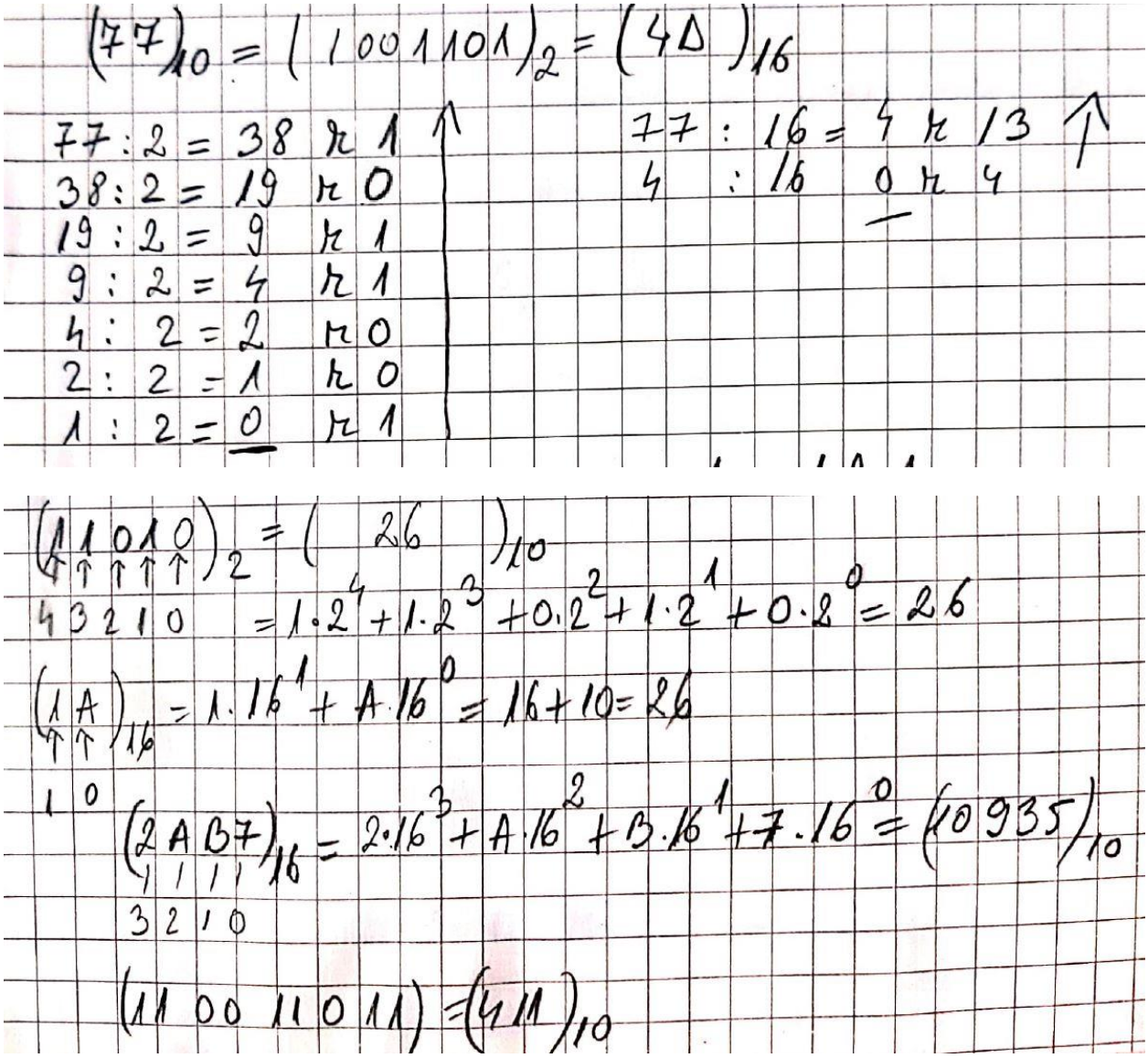
Conversii intre baze de numeratie

baze de numeratie:

baza 10 : 0 -> 9

baza 2 : 0 si 1

baza 16 : 0 -> 9, A, B, C, D, E, F



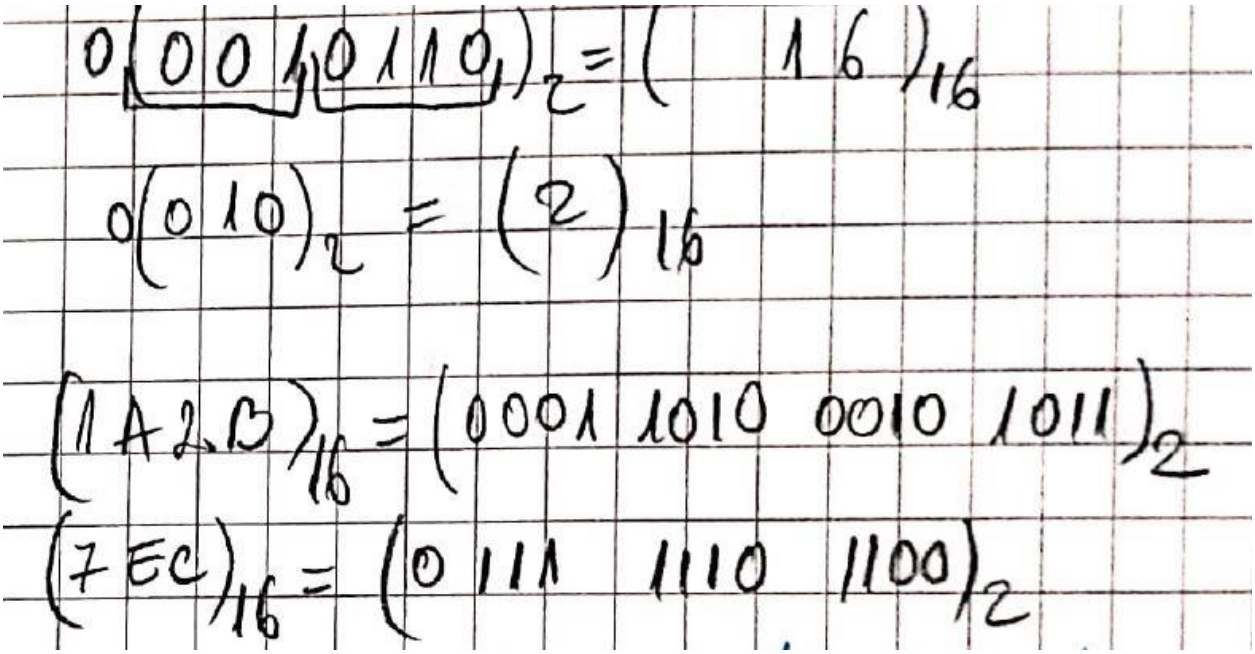
b2 -> b16

b 16 -> b2

Conversii rapide intre bazele 2 si 16

1 cifra hexazecimala = 1 grup de 4 cifre binare

Base 10	Base 16	Base 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111



Intrebari grila pentru conversii:

1. Care este valoarea in binar a numarului 101 din baza 10?

a. 1110 0101

b. 0110 0101

c. 0101 1010

d. 1010 0101

2. Care este valoarea numarului (0AB34)h in binar?

a. 0101 1010 1011 0011 0100

b. 1111.1010 1011 0011 0100

c. 1010 1011 0011 0100 0000

d. 0000 1010 1011 0011 0100

3. Care este valoarea numarului (11 0101)b in baza 10?

a. 53

b. 52

c. 54

d. 35

4. Care este valoarea numarului (11 0101)b in baza 16?

a. 35

b. 36

c. 34

d. 53

Problema

Enunt:

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 $S_n = X_1^n + X_2^n$, fara a se calcula radacinile ecuatiei X_1 si X_2.

Exemple:

Daca avem ecuatia: $X^2 – 6 * X + 8 = 0$ si n=2 => rezultatul este 20 (x1 =4 si x2=2)

Daca avem ecuatia: $X^2 – 8 * X + 15 = 0$ si n=2 => rezultatul este 34 (x1 = -3 si x2= -5)

Explicatii:

Stiind ca X1 si X2 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:

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

Ceea ce am putea scrie in mod echivalent:

$S_{n+2} – s*S_{n+1} + p*S_n = 0$ rezulta deci:

$S_{n+2} = s*S_{n+1} - p*S_n$

noi trebuie sa calculam: $S_n = X_1^n + X_2^n$

- $S_n = s*S_{n-1} - p*S_{n-2}$
- $S_n = X_1^n + X_2^n = s*S_{n-1} - p*S_{n-2}$

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

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

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;

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.
```

Probleme tip grilă

1.

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

a. (x<8) && (x>=5)

b. (x<=8) || (x>5)

c. (x>8) || (x<=5)

d. (x<=8) && (x>5)

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

a. (x%y==0) && (y%x==0) && (x*y>0)

b. (x<=y) && (y<x)

c. (x<=y) || (y<=x)

d. x*x==y*y

2. Se da urmatoarea secventa de cod scrisa in pseudocod.

a)

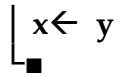
```
citeşte n (număr natural)
z ← 0
p ← 1
┌cât timp n>0 execută
│ c← n%10
│ n← [n/10]
│ ┌dacă c%3=0 atunci
│ │ z← z+p*(9-c)
│ │ p← p*10
│ └■
└■
scrie z
```

Care este rezultatul daca se executa pentru i) n = 103? ii) n = 103456?

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

b)

```
citeşte x (număr natural nenul)
┌cât timp x>0 execută
│ citeşte y (număr natural)
│ ┌dacă x>y atunci
│ │ scrie x%10
│ │altfel
│ │ scrie y%10
│ └■
```



Care se va afisa dupa executia codului daca se citesc valorile **17 22 13 101 2 7 5 0?**

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

c. Care este valoarea care se va returna la apelul: **rez = C (12345);**

```
int C (int n)
        {
                return ( n / 1000 ) * 100 +  n % 100 ;
    }
```

a. 1245 **b. 145** **c. 1345** **d. 12045** **e. 14500**

Raspunsuri grile:

1a - d; 1b - a; 2a - b; 2b – a; 2c-a

Problema

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

```
/*
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
*/
```

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 citeşte de la tastatură şase valori naturale nenule, apoi verifică, utilizând apeluri utile ale subprogramului de la subpunctul anterior dacă primele trei numere citite pot constitui laturile unui triunghi şi 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 puţin unul dintre cele două triplete de valori nu pot constitui laturile unui triunghi, programul va afişa pe ecran mesajul **nu se poate forma un triunghi** .

Analiza. Triunghiurile sunt congruente dacă există cele două triunghiuri şi 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, pana se verifica toate laturile).

Pentru a simplifica procesul de comparare se recomanda ca laturile sa fie ordanate

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))
    {
      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 parametrii
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 parametrii
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;

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.
```