Probleme consultații 20 ianuarie 2018

Algoritmi elementari

Problema 1: zerouri factorial

Enunt

Fiind dat un numar natural n, sa se determine numarul de cifre 0 cu care se termină factorialul numărului n:

a) prin calcularea factorialului
b) fără calcularea factorialului.

Programul care calculeaza factorialul

C++

```
/* Determină numarul de zerouri de la sfarsitul lui n!, calculand factorialul.
In: - n: numarul pentru care se determina zerourile terminale in n!.
Out: - numarul de zerouri terminale ale lui n!.
*/
int cifre0_1(intn)
{
	int f = factorial(n);
	int nr = 0;
	while (f % 10 == 0)
	{
		nr++; f /= 10;
	}
	return nr;
	   }

/* Determină factorialul unui numar. n! = 1*2*3*...*n.
In: - n: numarul pentru care se calculeaza factorialul.
Out: - factorialul lui n, n!.
*/
int factorial(intn)
{
	int f = 1;
	for (int i = 1; i <= n; i++)
	{
		f *= i;
	}
	return f;
}
```

Pascal

```
{Determină factorialul unui numar. n! = 1*2*3*...*n. In: - n: numarul pentru care se
calculeaza factorialul. Out: - factorialul lui n, n!. }

function factorial(n: integer):longword;
var f:longword;
i:integer;
begin
    f:=1;
```

```
    for i:=1 to n do
    f:=f*i;
    factorial1 := f;
end;

{ Determină numarul de zerouri de la sfarsitul lui n!, calculand factorialul. In: - n:
numarul pentru care se determina zerourile terminale in n!. Out: - numarul de zerouri
terminale ale lui n!. }
function cifre0_1(n:integer):integer;
var f:longword;
nr:integer;
begin
    f:=factorial1(n);
    nr:=0;
    while f mod 10 = 0 do
    begin
        inc(nr);
        f:=f div 10;
    end;
    cifre0_1:=nr;
end;
```

Programul care nu calculeaza factorialul

C++

```
/*
Determină numarul de zerouri de la sfarsitul lui n!, fara a calcula factorialul.
In:
- n: numarul pentru care se determina zerourile terminale in n!.
Out:
- numarul de zerouri terminale ale lui n!.
*/
int cifre0_2(intn) {
	int nr = 0;
	for (int i = 5; n / i >= 1; i *= 5) {
		nr += n / i;
	}
	return nr;
}
```

Pascal

```
{ Determină numarul de zerouri de la sfarsitul lui n!, fara a calcula
factorialul.
      In: - n: numarul pentru care se determina zerourile terminale in n!.
      Out: - numarul de zerouri terminale ale lui n!. }
function cifre0_2(n:integer):integer;
var nr,i:integer;
begin
    nr:=0;
    i:=5;
    while n div i >= 1 do
    begin
        nr:=nr + n div i;
        i:=i*5;
    end;
```

```
    cifre0_2:=nr;
end;
begin
  writeln('Hello, world!');
end.
```

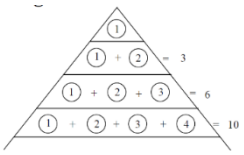
Problema 2: Numere triangulare

Enunt

Se citesc mai multe șiruri de numere naturale, fiecare șir se termină cu valoarea -1, iar citirea tuturor șirurilor se termină cu șirul vid. Se cere:

a) Pentru fiecare șir citit să se determine maximul dintre numerele triangulare din șir.

b) Maximul dintre numerele triangulare existente în toate șirurile citite.



Un număr x se numește triangular dacă există un număr n, astfel încât x este egal cu suma primelor n nr naturale.

C++

```
/*
Verifica daca un nr natural este triangular
In:
- x: nr natural dat
Out:
- true sau false, true daca x e piramidal, altfel false
*/

bool esteTriangular(intx) {
	int n = 1, s = 0;
	while (s <x) {
		s = s + n;
		n++;
	}
	if (s == x) returntrue;
	elsereturnfalse;
}

/*afiseaza maximul dintre numerele triangulare din fiecare sir citit, iar apoi
afiseaza maximul global
*/
void nrTriangulareMaximaleDinSir() {
	int n, max, maxTriangular = -1;
	cin >> n;
	while (n > -1) {
		max = -1;
		while (n > -1) {
			if (esteTriangular(n) && n > max)
				max = n;
			cin >> n;
		}
		if (max > maxTriangular) maxTriangular = max;
```

```
            cout <<"Maximul dintre numerele piramidale din sirul citit este "<< max;
            cin >> n;
      }
      cout <<"Maximul dintre toate numerele piramidale din toate sirurile este "<<
maxTriangular;
}
```

Pascal

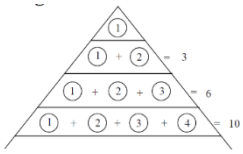
```
{
Verifica daca un nr natural este triangular
In:
- x: nr natural dat
Out:
- true sau false, true daca x e piramidal, altfel false
}

function esteTriangular(int x):boolean;
integer n,s;
begin
      n := 1; s := 0;
      while (s < x) do
      begin
            s: = s + n;
            n:=n+1;
      end;
      esteTriangular:=(s=x);
end;

{afiseaza maximul dintre numerele triangulare din fiecare sir citit, iar apoi
afiseaza maximul global
}
procedure nrTriangulareMaximaleDinSir();
var n, max, maxTriangular:integer;
begin
      maxTriangular = -1;
      readln(n);
      while (n > -1) do
      begin
            max := -1;
            while (n <> -1)
            begin
                  if (esteTriangular(n) and n > max)
                        max := n;
                  readln(n);
            }
            if (max > maxTriangular) maxTriangular := max;
            writeln("Maximul dintre numerele piramidale din sirul citit este "
,max);
            readln(n);
      }
      writeln("Maximul dintre toate numerele piramidale din toate sirurile este ",
maxTriangular);
}
begin
  writeln('Hello, world!');
```

```
end.
```

Problema 3: Numere piramidale



1
$1 + 3 = 4$
$1 + 3 + 6 = 10$
...

Enunt

Sa se scrie o functie care afiseaza primele k numere piramidale, unde k este un nr natural specificat ca parametru functiei.

Un număr x se numește piramidal daca exista un numar natural n astfel incat x poate fi scris ca suma primelor n numere triangulare.

C++

```
/*afiseaza primele k numere piramidale
IN: k – nr natural
OUT: -
*/
void sirPiramidal(intk) {
	int s = 1, n = 2, nrPiramidal = 1;
	for (int i = 0; i <k; i++) {
		cout << nrPiramidal <<" ";
		s = s + n;
		n = n + 1;
		nrPiramidal = nrPiramidal + s;
	}
}
```

Pascal

```
{afiseaza primele k numere piramidale
IN: k – nr natural
OUT: -
}
procedure sirPiramidal(k:integer);
var s,n,nrPir,i:integer;
begin
	s := 1; n:= 2; nrPir := 1;
	for i:=0 to k do
	begin
		writeln(nrPir);
		s := s + n;
		n := n + 1;
		nrPir := nrPir + s;
	end;
end;

begin
  sirPiramidal(3);
end.
```

Pentru k=8 se va afisa: 1 4 10 20 35 56 84 120 165

Problema 4: Numar de prefixe permutari

Enunt

Se citeste un numar n si apoi un sir de n numere naturale reprezentand o permutare a multimii {1,2,3,...n}. Afisati cate dintre prefixele sirului citit sunt la randul lor permutari.
Exemplu:
12
2 1 7 3 4 5 8 6 9 12 10 11

4
Explicatie:
Cele patru permutari prefix sunt:
2 1
2 1 7 3 4 5 8 6
2 1 7 3 4 5 8 6 9
2 1 7 3 4 5 8 6 9 12 10 11

C++

```
int n, x, s = 0, k = 0;
int main()
{
	cin >> n;
	for (int i = 1; i <= n; i++)
	{
		cin >> x;
		s = s + x;//fac suma pana la i
		if (s == i*(i + 1) / 2) //daca e suma de 1+2+..+i, atunci e prefix
permutare
			k++;
	}
	cout << k;
	return 0;
}
```

Pascal

```
var n,x,s,k,i:integer;
begin
  s:=0; k:=0;
  readln(n);
  for i:=1 to n do
  begin
	readln(x);
	s := s+x;
	if (s = (i*(i+1) div 2)) then k:=k+1;
  end;
  write(k);
end.
```

Problema 5: secvențe 101

Enunț

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). Pentru un șir s citit de la tastatură să se determine grupul de numere de grad k.

C++

O posibilă rezolvare poate utiliza un vector de 32 de elemente (daca elementele șirului sunt numere naturale de tipul intreg reprezentarea binară a unui intreg are 32 de biți, dimensiunea unui intreg este 4 octeți).

```
#include <iostream>
using namespace std;
const int NRBITI = 32;
int BINAR[NRBITI] = {0};

/*
 * toate elementele vectorului BINAR = 0
 */
void resetBinar(){
    for (int i = 0; i < NRBITI; ++i) {
        BINAR[i] = 0;
    }
}

/*
 * 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;
}

/*
 * 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 putin 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 putin " << k << " :" << endl;
    for (int y = 0; y < i; ++y) {
        cout << s[y] << " ";
    }
    cout << endl;
    return 0;
}
```

Exemple:

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

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 $(101)_2 = (5)_{10}$).

```
#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
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

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;
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
            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.
```

Exemple:

Date de intrare	Rezultat
3 362 360 85 0 362 360 85 0	numarul 362 are gradul= 3 numarul 85 are gradul= 3 numere de grad 3 : 362 85

Problema 6: numere zâmbărețe

Enunț:

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 citeste mai multe numere pana la citirea numarului 0 si determină câte numere "zâmbărețe" s-au citit.

Exemplu:

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

C++

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

/*
 * 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);
}

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;
}
```

Exemple:

Date de intrare si rezultat
23 numarul 23 este zambaret 45 56 46 49 numarul 49 este zambaret 19 numarul 19 este zambaret 100 numarul 100 este zambaret 11 0 Am citit 4 numere zambarete

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 cu n
    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;
    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 19 100 11 0	23 este zambaret 49 este zambaret 19 este zambaret 100 este zambaret Am citit 4 numere zambarete

Probleme tip grilă

1. Ce va afișa următoarea secvență de program, dacă se citesc în această ordine numerele: 194, 121, 782?

```
citește x (număr natural)
```

```
n=0
┌cât timp x≠0 execută
│ y=x; c=0
│ ┌cât timp y>0 execută
│ │ ┌dacă y%10>c atunci
│ │ │ c=y%10
│ │ └■
│ │ y=y/10
│ └■
│ n=n*10+c
│ citește x
└■
scrie n
```

Raspuns: 928

2. Care este valoarea afişată pentru **x=-2** şi **m=9**?

```
citeşte x ,m
{x întreg, m natural}
y←1
┌ cât timp m>0 exec
│┌ dacă m%2 =0
│| atunci
│|   m←[m/2]; x←x*x
│| altfel
│|   m←m-1;y←y*x
│└■
└■
scrie y
```

Raspuns:-512

3. Ce face secvența:

```
int f(char* str)
{
    if (*str == '\0')
        return 0;
    else
     return 1 + f(str + 1);
}
```

Raspuns: calculeaza lungimea unui șir de caractere

4. Ce face secvența:

```
int f(int n) {
    if (n == 0)
        return 0;
    else
        return (n % 2 + 10 * f(n / 2));
}
```

Raspuns: transformă un numar în baza doi