23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Algoritmi Tablouri Unidimensionale

Problema 1 ~ *Joc: Păsărică mută-ți cuibul*

Enunț

Se dau n (n<=60) cuiburi pe care se află n-1 copii astfel încât fiecare copil se află într-un cuib. În fiecare moment orice copil se poate muta în cuibul liber.

Se dă poziţia (*configuraţia*) inițială și poziţia finală a copiiilor. Se cere să se determine poziţiile intermediare (ale copiiilor) după fiecare schimbare a cuibului (*mutare*), plecând de la *configuraţia* inițială până la *configuraţia* finală. Obs. Cuibul în care nu se afla nici un copil este notat cu 0.

- **Exemplul 1. :**
  n=4
  Configuratia initială: **3 2 0 1**
  Configuratia finală: **2 1 3 0**
  Soluţie:

		1	2	3	4
		3	**2**	**0**	**1**
Pasul 1.	Copilul 3 pleacă de la cuibul 1 la cuibul 3:	0	2	3	1
Pasul 2.	Copilul 2 pleacă de la cuibul 2 la cuibul 1:	2	0	3	1
Pasul 3.	Copilul 1 pleacă de la cuibul 4 la cuibul 2:	2	1	3	0
		2	**1**	**3**	**0**

Analiză

- Se va porni de la configuratia initiala si pentru fiecare pozitie i a caror elemente nu corespund in cele doua configuratii, se va proceda in doi pasi astfel:
  - Pasul a) se va elibera spatiu in configuratia initiala, respectiv 0 pe pozitie
  - Pasul b) se va cauta pozitia din cofiguratia initiala pe care se gaseste elementul din configuratia finala de pe pozitia i

- **Exemplul 2. :**

Configuraţia initială: **2, 0, 3, 4, 5, 1.**
Configuraţia finală: **5, 4, 1, 0, 3, 2.**

Mutări:

* Start * >>> Config. : **2, 0, 3, 4, 5, 1.**
1 <-_-> 2 >>> Config. : 0, 2, 3, 4, 5, 1.
1 <-_-> 5 >>> Config. : 5, 2, 3, 4, 0, 1.
2 <-_-> 5 >>> Config. : 5, 0, 3, 4, 2, 1.
2 <-_-> 4 >>> Config. : 5, 4, 3, 0, 2, 1.
3 <-_-> 4 >>> Config. : 5, 4, 0, 3, 2, 1.
3 <-_-> 6 >>> Config. : 5, 4, 1, 3, 2, 0.
4 <-_-> 6 >>> Config. : 5, 4, 1, 0, 2, 3.
5 <-_-> 4 >>> Config. : 5, 4, 1, 2, 0, 3.
5 <-_-> 6 >>> Config. : 5, 4, 1, 2, 3, 0.
6 <-_-> 4 >>> Config. : **5, 4, 1, 0, 3, 2.**
* Stop *

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Specificarea funcțiilor

Subalgoritmul **citesteConfiguratii (n,ci,cf)**:
Descriere: Citeste numarul de cuiburi, configuratia initiala si configuratia finala
Date:
Rezultate: n – numarul de cuiburi,ci, cf –vectorul de configuratie initiala, respectiv finala.

Subalgoritmul **tiparesteConfiguratie (n,x)**:
Descriere: Tipareste o configuratie a celor n cuiburi
Date: n- numarul de cuiburi, x - configuratia ce se tipareste
Rezultate: -.

Subalgoritmul **TiparesteMutare (pozi, pozf, n,config)**:
Descriere: Tipareste o mutare in configuratia config
Date: n- numarul de cuiburi, config - configuratia ce se tipareste, pozi,pozf -cuiburile la care s-au schimbat valorile
Rezultate: -.

Functia **swap (x,y)**:
Descriere: Interschimba valorilor celor doi parametri
Date: x,y-valori intregi
Rezultate: x,y- valorile interschimbate a celor doi parametri

Subalgoritmul **schimbaCuib (n,ci,cf,poz)**:
Descriere: Determina obtinerea valorii cf[poz] pe pozitia poz din configuratia ci
Date: n- numarul de cuiburi, ci, cf -configuratii, poz-pozitia pe care se incearca obtinerea valorii din configuratia finala
Rezultate: ci - configuratia initiala modificata.

Functia **cautaValoare (n,x,val)**:
Descriere: Determina pozitia pe care apare valoare *val* in vectorul x cu n elemente.
Date: n – lungime vector , x –vectorul (elementele incep de pe pozitia 1), val - valoarea cautata.
Rezultate: 0 - daca val nu apare in vectorul x
poz - cu proprietatea ca x[poz]=val;

Subalgoritmul **determinaMutari (n,ci,cf)**:
Descriere: Determina mutarile de la configuratie initiala *ci* la configuratia finala *cf*. Configuratiile intermediare sunt afisate pe masura determinarii lor
Date: n- numarul de cuiburi, ci, cf -configuratia initiala, respectiv finala
Rezultate: -

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Implementare

Varianta C++ — Varianta a)

```
#include<iostream>
using namespace std;
#define MAX 50
typedef int sir[MAX];
int cautaValoare(int n, sir x, int val){
        for(int i=1;i<=n;i++)
                if (x[i]==val)   return i;
        return 0;
}
void citesteConfiguratii(int &n, sir ci, sir cf){
        cout<<"n="; cin>>n;
        cout<<"Configuratia initiala :"<<endl;  for (int i=1;i<=n;i++)  cin>>ci[i];
        cout<<"Configuratia finala: "<<endl;    for (int i=1;i<=n;i++)  cin>>cf[i];
}
void tiparesteConfiguratie(int n, sir x){
        for(int i=1; i<=n; i++)
                cout<<x[i]<<" ";
        cout<<endl;
}
void swap(int &x, int &y){
        int z=x;      x=y;       y=z;
}
void TiparesteMutare(int pozi,int pozf, int n, sir conf){
        cout<<"Interschimbare "<<pozi<<" cu "<<pozf<<"--> ";
        tiparesteConfiguratie(n,conf);
}
void schimbaCuib(int n, sir ci, sir cf, int poz){
        if (ci[poz]!=0){
                int pozZero=cautaValoare(n,ci,0);
                swap(ci[poz],ci[pozZero]);
                TiparesteMutare(poz,pozZero,n,ci);       }
        if (ci[poz]!=cf[poz]){
                int pozCf=cautaValoare(n,ci,cf[poz]);
                swap(ci[poz], ci[pozCf]);
                TiparesteMutare(poz,pozCf,n,ci);         }
}
void determinaMutari(int n, sir ci, sir cf){
        for(int i=1;i<=n;i++)
                if (ci[i]!=cf[i])    schimbaCuib(n,ci,cf, i);
}
int main(){
        int n;
        sir ci,cf;
        citesteConfiguratii(n,ci,cf);
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
        determinaMutari(n,ci,cf);

}
```

Varianta C++ Varianta b)

```
#include <iostream>
#include <fstream>

using namespace std;

void  Citeste(int* Ci, int* Cf, int& n)        // Citeste Config. Init. si Finala
{
      ifstream Cin("Pasarica Muta Cuibul.Txt");
      Cin >> n;
      for (int i=1; i<=n; i++)  Cin >> Ci[i];
      for (int i=1; i<=n; i++)  Cin >> Cf[i];
      Cin.close();
}

void  Tipareste(int* Config, int n)             // Tipareste o Config.
{
      cout << "  >>>  Config. : ";
      for (int i=1; i<=n; i++)
           cout << Config[i] << ", ";
      cout << "\b\b.  \n";
}

void  Swap(int& x, int& y)  { int z=x; x=y, y=z; }    // Interschimba doua elem.: x,y

void  Swap(int& ci, int cf, int* Ci, int* Cf, int* Poz) // Swap ci <-> cf in Conf. init.
{
      int   cv=ci;
      Swap(    ci ,Ci[Poz[cf]]),
      Swap(Poz[ci],   Poz[cv] );
}

void  Schimba_C(int* Ci, int* Cf, int* Pf, int n)  // Config. Init. -> Config. Finala
{
      int Poz[n], k=0;
      for (int i=1; i<=n; i++) Poz[Ci[i]]=i;       // Memoreaza pozitiile initiale
      for (int i=1; i<=n; i++)                     // Completeaza pozitia i
           if (Ci[i]-Cf[i])                        // Poz. i nu este corecta ?
           if (Ci[i]*Cf[i])                        // Este libera (0)?
               Pf[k++]=Poz[0],                Swap(Ci[i],     0,    Ci,Cf,Poz),
               Pf[k++]=Poz[Cf[i]],            Swap(Ci[Poz[0]],Cf[i],Ci,Cf,Poz); else
               Pf[k++]=Poz[Cf[i]], Pf[k++]=0, Swap(Ci[i],     Cf[i],Ci,Cf,Poz); else
               Pf[k++]=            Pf[k++]=0;

}

void  MutaPasarica(int* Cc, int* Pf, int i, int& k, int n)      //Executa mutarea k
{
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
48        if (Pf[k++]) Swap(Cc[i],Cc[Pf[k-1]]),                    //Mutare nenula
49                     cout << "\n " << i << " <-_-> " << Pf[k-1],
50                     Tipareste(Cc,n);
51  }
52
53  void  Print_Mut(int* Cc, int* Pf, int n)    //Tipareste mutarile si configuratiile succesiv
54  {                                  int k=0;
55        for (int m=1; m<=2*n; m++)
56             MutaPasarica(Cc,Pf,(m+1)/2,k,n); // Executa mutarea m, pune corect poz.  i=(m+1)/2
57  }
58
59  void  Copy(int* Ci, int* Cc, int i)         // Salveaza pozitia initiala:  Cc=Copie(Ci)
60  {
61        if (i) Copy(Ci,Cc,i-1), Cc[i]=Ci[i];
62  }
63
64  int   main()
65  {
66        int       Ci[50],Cf[50],  n;
67        Citeste  (Ci,    Cf,      n);  int Pf[2*n], Cc[n];  cout << "\n * Start *";
68        Tipareste(Ci,             n);  Copy(Ci,Cc,n);
69        Schimba_C(Cc,    Cf, Pf,  n);
70        Print_Mut(Ci,        Pf,  n);  cout << "\n * Stop * \n";
71  }
```

Implementare

Varianta Pascal

```
Program Cuiburi;
type vector=array[1..100] of integer;
procedure citesteConfiguratii(var n:integer; var ci,cf:vector);
var i:integer;
begin
   writeln('Introduceti nr. de cuiburi');
   readln(n);
   writeln('Introduceti configuratia initiala');
   for i:=1 to n do
     begin
       read(ci[i]);
     end;
    writeln('Introduceti configuratia finala');
   for i:=1 to n do
     begin
       read(cf[i]);
     end;
end;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
Procedure swap(var x,y:integer);
var aux:integer;
begin
   aux:=x;
   x:=y;
   y:=aux;
end;

function cautaValoare(n:integer; x:vector; val:integer):integer;
var i,poz:integer;
begin
   i:=1;
   poz:=-1;

   while (i<=n) and (poz<0) do
     begin
        if (x[i]=val) then poz:=i;
        inc(i);
    end;
    cautaValoare:=poz;
end;

procedure TiparesteConfiguratie(n:integer; config:vector);
var i:integer;
begin
  for i:=1 to n do
     write(config[i],' ');
  writeln;
end;

Procedure TiparesteMutare(pozi,pozf:integer; n:integer; config:vector);
begin
  write('Interschimbare ',pozi,' cu ',pozf,' -->');
  TiparesteConfiguratie(n,config);
end;

procedure schimbaCuib(n:integer; var configInit:vector; configFin:vector;
poz:integer);
var pozZero, pozFinal:integer;
begin
   if (configInit[poz]<>0) then
     begin
         pozZero:=cautaValoare(n,configInit,0);
         swap(configInit[poz], configInit[pozZero]);
         TiparesteMutare(poz,pozZero,n,configInit);
     end;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
    if (configInit[poz]<>configFin[poz]) then
     begin
        pozFinal:=cautaValoare(n, configInit, configFin[poz]);
        swap(configInit[poz], configInit[pozFinal]);
        TiparesteMutare(poz, pozFinal,n, configInit);
     end;
end;


procedure DeterminaMutari(n:integer; configInit,configFinal:vector);
var i:integer;
begin
  for i:=1 to n do
     if (configInit[i]<>configFinal[i]) then
         schimbaCuib(n,configInit, configFinal,i);
end;

var  n:integer;
     ci,cf:vector;
begin
  citesteConfiguratii(n,ci,cf);
  DeterminaMutari(n,ci,cf);
end.
```

Problema 2:

Fiind dat un șir $X=(x_1, x_2, \ldots, x_n)$ de numere naturale nenule ($1<=xi<=3000$, $1<=n<=500$), să se determine poziția de început și lungimea celei mai lungi subsecvențe de numere prime din șir.

Exemple:

X=(2, 3, 5, 7) => pozitia=1, lungimea=4, secventa (2,3,5,7)

X=(4, 14, 24, 34) => lungimea=0, pozitia ?

X=(2, 7, 11, 15, 17, 23, 29) => pozitia=1, lungimea=3, secventa (2,7,11)

Sau pozitia=4, lungimea=3, secventa (17,23,29)

Analiză

Se parcurge sirul X si pentru pozitia curenta *i* se obtine lungimea maxima a subsecventei de numere prime care incepe pe pozitia *i.* Daca lungimea obtinuta, este mai mare decat lungimea maxima obtinuta anterior se retine noua lungime si pozitia de inceput.

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Specificarea funcțiilor

Subalgoritmul **citireSir(x,n)**
Descriere: citeste sirul x cu n elemente
Date: -
Rezultate: n- numarul de elemente, x-elementele sirului

Functia **prim(x)**
Descriere: determina daca numarul *x* este prim
Date: x
Rezultate: true - daca x este prim,
false - alfel

Functia **determinaSecvPrime(x, n, i)**
Descriere: determina lungimea maxima a subsecventei de numere prime din sirul *x* care incepe pe pozitia *i*
Date: x- elementele sirului, n - numarul de elemente a sirului x, i -pozitia de inceput
Rezultate: 0- daca x_i nu e numar prim
l - lungimea secventei (x_i, x_{i+1}, ...x_{i+l-1} sunt numere prime)

Subalgoritmul **DeterminaSecventaMaxima(x, n, poz_start, lungime)**
Descriere: determina pozitia de inceput si lungimea subsecventei maxime de numere prime din sirul x
Date: x- elementele sirului, n - numarul de elemente din x
Rezultate: poz_start - pozitia de inceput
lungime - lungimea maxima, sau 0 daca nu exista numere prime in sirul x

Subalgoritmul **tiparireSecv(x, poz_start, lungime)**
Descriere: tipareste o subsecventa a sirului **x**, subsecventa incepe pe pozitia **poz_start** si contine **lungime** elemente
Date: x- elementele sirului, poz_start-pozitia de inceput, lungime -nr. de elemente din subsecventa
Rezultate:

Implementare

Varianta C++

```
#include<iostream>
using namespace std;
#define MAX 500
typedef int sir[MAX];
void citireSir(sir x, int&n){
        cout<<"Introduceti numarul de elemente ";
        cin>>n;
        for(int i=1; i<=n;i++){
                cout<<"x["<<i<<"]=";
                cin>>x[i];
        }
}
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
bool prim(int x){
        if (x<2)
                return false;
        for(int i=2;i*i<=x;i++)
                if (x%i==0)
                        return false;
        return true;

}

int determinaSecvPrime(sir x, int n, int i){
        int l=0;
        while((i+l<=n)&&prim(x[i+l]))
                l++;
        return l;
}

void DeterminaSecventaMaxima(sir x, int n, int &poz_start, int &lungime){
        poz_start=0;
        lungime=0;
        int i=1, lung_secv;
        while (i<=n) {
                lung_secv=determinaSecvPrime(x,n,i);
                if (lung_secv>lungime){
                        lungime=lung_secv;
                        poz_start=i;
                }
                i=i+lung_secv+1;
        }
}

void tiparireSecv(sir x, int poz_start, int lung){
        if (lung==0)
                cout<<"Sevcenta vida"<<endl;
        else{
                cout<<"Lungimea "<<lung<<" pozitia"<<poz_start<<endl;
                for(int i=poz_start; i<poz_start+lung;i++)
                        cout<<x[i]<<' ';
                cout<<endl;
        }

}
int main(){
        sir x;
        int n,lung,pozstart;
        citireSir(x,n);
        DeterminaSecventaMaxima(x,n,pozstart,lung);
        tiparireSecv(x,pozstart,lung);

}
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Implementare

Varianta Pascal

```
Program SecventeMaximePrime;
type vector=array[1..100] of integer;

procedure citireSir(var n:integer; var x:vector);
var i:integer;
begin
   writeln('Introduceti nr. de elemente');
   readln(n);
   writeln('Introduceti elementele');
   for i:=1 to n do
     begin
       read(x[i]);
     end;
end;
Function prim(x:integer):boolean;
var   d:integer;
begin
      if (x<3 or x Mod 2=0) then  prim:=x=2 else     begin
      d:=3;
      while (d*d <= x) and (x Mod d > 0) do d:=d+2;
      prim:= d*d >  x;                               end;
end;
Function DeterminaSecvPrim(n:integer;x:vector;poz:integer):integer;
var l:integer;
begin
  l:=0;
  while (poz+l<=n) AND prim(x[poz+l]) do inc(l);
  DeterminaSecvPrim:=l;
end;
Procedure DeterminaSecventaMaxima(n:integer;x:vector; var start,lungime:integer);
var i,lung_secv:integer;
begin
start:=0;
lungime:=0;
i:=1;
while i<=n do
  begin
     lung_secv:=DeterminaSecvPrim(n,x,i);
     if (lung_secv>lungime) then
        begin
           lungime:=lung_secv;
           start:=i;
        end;
     i:=i+lung_secv+1
  end;
end;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
Procedure TiparireSecv(n:integer; x:vector; start,lungime:integer);
var i:integer;
begin
if lungime=0 then
    writeln('secvventa vida')
else
   begin
     writeln('Secventa are lungimea ',lungime, ' si incepe la poz ',start);
     for i:=start to start+lungime-1 do
          write(x[i], ' ');
     writeln;
   end;
end;
var n,start,lung:integer;
     x:vector;
begin
  citireSir(n,x);
   DeterminaSecventaMaxima(n,x,start,lung);
  TiparireSecv(n,x,start,lung)
end.
```

Problema 3:

Fiind dat un șir $X=(x_1, x_2, \ldots, x_n)$ de numere naturale nenule ($1<=x_i<=3000$, $1<=n<=500$), să se determine pozițiile de început și lungimea tuturor subsecvențelor maximale de numere *prime între ele* din șir.

Exemple:
X=(2, 2, 2) => lungimea=0, poziția ?
X=(2, 3, 5) => lungimea=3, poziția=1
X=(20, 21, 121) => lungimea=3, poziția=1
X=(4, 2, 3, 8, 11, 13, 17, 9, 23, 29, 31, 20) => lungimea = 8, pozițiile= [4, 5]
X=(4, 2, 3, 8, 11, 13, 17, 9, 23, 29, 31, 21) => lungimea = 8, poziția=4

Analiză

Se parcurge sirul X si pentru fiecare poziti *i* se obtine lungimea maxima a subsecventei de numere prime intre ele care incepe pe pozitia *i.* Daca lungimea obtinuta, este mai mare decat lungimea maxima obtinuta anterior se retine noua lungime si pozitia de inceput. daca lungimea obtinuta este egala cu lungimea obtinuta anterior, se adauga la sirul pozitiilor de inceput.

Specificarea funcțiilor

Subalgoritmul **citireSir(x,n)**
Descriere: citeste sirul x cu n elemente
Date: -
Rezultate: n- numarul de elemente, x-elementele sirului

Functia **cmmdc(a, b)**
Descriere: determina cel mai mare divizor comun al numerelor a si b

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Date: a, b (a>0, b>0)
Rezultate: cmmdc(a,b)

Functia **DeterminaLungSecvPrimeIntreEle(x, n, i)**
Descriere: determina lungimea maxima a subsecventei de numere prime intre ele din sirul *x* care incepe pe pozitia *i*
Date: x- elementele sirului, n - numarul de elemente a sirului x, i -pozitia de inceput
Rezultate: l - lungimea secventei, sau 0

Subalgoritmul **DeterminaSecventeMaxime(x, n, lungime, pozitii, mpoz)**
Descriere: determina pozitiile de inceput si lungimea subsecventelor maximale de numere prime intre ele din sirul x
Date: x- elementele sirului, n - numarul de elemente din x
Rezultate: pozitii- pozitia de inceput
lungime - lungimea maxima, sau 0 daca nu exista numere prime intre ele in sirul x
mpoz-numarul de secvente gasite

Subalgoritmul **TiparireSecv(x,lungime, start, mpoz)**;
Descriere: Tipareste toate subsecventele cu proprietatea ceruta
Date: x - elementele sirului, start - pozitiile de inceput alte subsecventelor cerute, mpoz- numarul de subsecvente, lungime- lungimea unei subsecvente
Rezultate: -

Implementare

Varianta C++

```
#include<iostream>
using namespace std;
#define MAX 500
typedef int sir[MAX];

void citireSir(sir x, int&n){
	cout<<"Introduceti numarul de elemente ";
	cin>>n;
	for(int i=1; i<=n;i++){
		cout<<"x["<<i<<"]=";
		cin>>x[i];
	}
}

//Preconditie a>0,b>0
int cmmdc(int a, int b){
	while(a!=b){
		if (a>b)
			a=a-b;
		else
			b=b-a;
	}
```

23 noiembrie 2019 Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
	return a;
}

int DeterminaLungSecvPrimeIntreEle(sir x, int n, int i){
	int l=1;
	bool stop=false;
	while((i+l<=n)&&(!stop))
	{
		stop=false;
		for(int k=i; (k<i+l)&&(!stop);k++)
			if (cmmdc(x[k],x[i+l])!=1)
				stop=true;
		if (!stop)
			l++;
	}
	return l;
}
void DeterminaSecventeMaxime(sir x, int n, int &lungime_max,sir rezultat, int &m){
	lungime_max=0;
	int i=1, lung_secv;
	m=0;
	while (i<=n) {
		lung_secv=DeterminaLungSecvPrimeIntreEle(x,n,i);
		if (lung_secv>1){
			if (lung_secv>lungime_max){
				lungime_max=lung_secv;
				m=1;
				rezultat[m]=i;

			}else{
				if (lung_secv==lungime_max)
				rezultat[++m]=i;
			}
		}
		i=i+1;
	}
}

void tiparireSecvente(sir x, int lung, sir pozitii, int nr_pozitii){
	if (lung==0)
		cout<<"Nu exista nici o secventa cu proprietatea ceruta"<<endl;
	else{
		int i=1;
		cout<<"Sunt "<<nr_pozitii<<" secvente de lungime maxima"<<lung<<endl;
		while(i<=nr_pozitii){
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
                    cout<<"Secventa "<<i<<endl;
                    for(int j=pozitii[i];j<pozitii[i]+lung;j++)
                          cout<<x[j]<<' ';
                    cout<<endl;
                    i++;
              }
       }
}
int main(){
       sir x,pozitii;
       int n,lung,pozstart,m;
       citireSir(x,n);
       DeterminaSecventeMaxime(x,n,lung,pozitii,m);
       tiparireSecvente(x,lung,pozitii,m);
}
```

Implementare

Varianta Pascal

```
Program SecventeMaximePrimeIntreEle;
type vector=array[1..100] of integer;

procedure citireSir(var n:integer; var x:vector);
var i:integer;
begin
   writeln('Introduceti nr. de elemente');
   readln(n);
   writeln('Introduceti elementele');
   for i:=1 to n do
     begin
       read(x[i]);
     end;
end;
Function cmmdc(a,b:integer):integer;
begin
 if (a>0) AND (b>0) then
   begin
      while (a<>b) do
        if (a>b) then a:=a-b
                 else b:=b-a;
      cmmdc:=a;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
   end
  else
     cmmdc:=a+b;
end;
Function DeterminaLungSecvPrimeIntreEle(n:integer;x:vector;poz:integer):integer;
var l,k:integer;
    stop:boolean;
begin
  l:=1;
  stop:=false;
  while (poz+l<=n) and (not stop) do
     begin
       stop:=false;
       k:=poz;
       while (k<poz+l) and (not stop) do
         begin
           if (cmmdc(x[k],x[poz+l])<>1) then stop:=true;

           inc(k);
         end;
         if (not stop) then inc(l);
     end;
  DeterminaLungSecvPrimeIntreEle:=l;
end;

Procedure DeterminaSecventeMaxime(n:integer;x:vector; var pozitii:vector;var mpoz,
lungime: integer);
var i,lung_secv:integer;
begin
mpoz:=0;
lungime:=0;
i:=1;
while i<=n do
  begin
     lung_secv:=DeterminaLungSecvPrimeIntreEle(n,x,i);
     if (lung_secv>lungime) then
        begin
            mpoz:=1;
           lungime:=lung_secv;
           pozitii[mpoz]:=i;
        end
    else
       if (lung_secv=lungime) then
          begin
            inc(mpoz);
            pozitii[mpoz]:=i;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
          end;
     i:=i+1
  end;
end;

Procedure TiparireSecv( x:vector; lungime:integer;start:vector; mpoz:integer);
var i,j:integer;
begin
if lungime=0 then
    writeln('secventa vida')
else
   begin
     Writeln('S-au gasit ',mpoz, ' secvente avand lungimea ',lungime);
     for j:=1 to mpoz do
       begin
            writeln('Secventa ',j, ' incepe la poz ',start[j]);
            for i:=start[j] to start[j]+lungime-1 do
                begin
                    write(x[i], ' ');
                end;
            writeln;
        end;
   end;
end;

var n,start,lung,mpoz:integer;
     x,poz:vector;
begin

  citireSir(n,x);
  {tiparireSir(n,x);}
  DeterminaSecventeMaxime(n,x,poz,mpoz,lung);
  TiparireSecv(x,lung,poz,mpoz);
end.
```

Problema 4:

Se consideră șirurile ***a*** cu ***n*** elemente (1 ≤ ***n*** ≤ 10 000) şi ***b*** cu ***m*** elemente (1 ≤ ***m*** ≤ 10 000) care sunt numere naturale mai mici decât 30 000. Spunem că șirul ***a*** *„se poate reduce”* la șirul ***b*** dacă există o împărţire a șirului ***a*** în subsecvenţe (o subsecvență conține unul sau mai multe elemente) disjuncte de elemente aflate pe poziţii consecutive în șirul ***a*** astfel încât prin înlocuirea fiecărei subsecvenţe cu suma elementelor sale să se obţină, în ordine, elementele șirului ***b***.
Scrieți un subalgoritm care stabilește dacă șirul ***a*** *se poate reduce sau nu* la șirul ***b***. În caz afirmativ, determinați poziția ***k*** în șirul ***b*** unde se află valoarea care se obține însumând cele mai multe elemente din șirul ***a***. Subalgoritmul are ca parametri de intrare cele două numere ***n*** și ***m***, precum și cele două șiruri ***a*** și ***b***. Parametrii de ieșire vor fi ***răspuns***, ***k*** și ***nrMax***, unde ***răspuns*** va avea valoarea ***adevărat*** dacă

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

răspunsul la întrebare este *DA*, respectiv ***fals***, în caz contrar, ***k*** reprezintă indicele elementului din șirul ***b*** care se obține însumând ***nrMax*** elemente din șirul ***a.***

***Exemplul* 1:**

n = 12, ***a*** = (7, 3, 4, 1, 6, 4, 6, 9, 7, 1, 8, 7),
m = 4 și ***b*** = (14, 7, 26, 16),

răspuns = ***adevărat***, deoarece 7 + 3 + 4 = 14, 1 + 6 = 7, 4 + 6 + 9 + 7 = 26, 1 + 8 + 7 = 16. Astfel, ***k*** = 3 și ***nrMax*** = 4.

***Exemplul* 2:**

n = 10, ***a*** = (7, 3, 1, 6, 4, 6, 9, 7, 1, 8),
m = 4 și ***b*** = (14, 7, 26, 16),

răspuns va avea valoarea ***false***, deoarece 7 + 3 + 1 = 11 < 14, iar 7 + 3 + 1 + 6 = 17 > 14, deci valoarea b_1 = 14 nu poate fi obținută însumând elemente consecutive din șirul ***a***.

În exemple șirurile sunt indexate începând cu 1.

Analiză

Se parcurg sirurile **a** si **b**. Fie ***ia-*** pozitia curenta din sirul ***a*** si ***ib*** pozitia curente din sirul ***b.*** Pentru elementul **b[ib]** determinam lungimea secventei din sirul **a**, care incepe pe pozitia **ia** si care are proprietatea ca suma tuturor elementelor din secventa este egala cu **b[ib].** Daca nu gasim o astfel de secventa, atunci sirul **a** nu se poate reduce la sirul **b**. Daca gasim o astfel de secventa **ia** se incrementeaza cu numarul de elemente din secventa gasita, iar **ib** se muta pe urmatorul element din sirul **b**.

Specificarea funcțiilor

Subalgoritmul **citireSir(x,n)**
Descriere: citeste sirul x cu n elemente
Date: -
Rezultate: n- numarul de elemente, x-elementele sirului

Functia **determinaSecv(a, n, ia, s)**
Descriere: determina lungimea secventei din sirul x care incepe pe pozitia ia cu proprietatea ca suma tuturor elementelor sale este egala cu s.
Date: a-elementele sirului, n- numarul de elemente din a, ia -pozitia de inceput a secventei, s-valoarea sumei
Rezultate: l- lungimea secventei
0- daca nu exista o secventa cu proprietatea ceruta

Subalgoritmul **reduce(a, n, b, m, posibil, k, nrMax)**

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Descriere: verifica daca sirul a poate fi redus la sirul b, conform cerintelor din enuntul problemei
Date: a- elementele sirului a, n- numarul de elemente din a, b- elementele sirului b, m-numarul de elemente din b
Rezultate: posibil -true daca a se poate reduce la b
-false, altfel
k, nrMax- (daca posibil este adevarat) k este indicele elementului din șirul ***b*** care se obține însumând ***nrMax*** elemente din șirul ***a***

Implementare

Varianta C++

```
#include<iostream>
using namespace std;
#define MAX 500
typedef int sir[MAX];

void citireSir(sir x, int&n){
	cout<<"Introduceti numarul de elemente ";
	cin>>n;
	for(int i=1; i<=n;i++){
		cout<<"x["<<i<<"]=";
		cin>>x[i];
	}
}

int determinaSecv(sir a, int n, int ia, int s){
	int lung=0;
	int sum=0;
	while((ia+lung<=n)&&(sum<s)){
		sum+=a[ia+lung];
		lung++;
	}
	if (sum==s)
		return lung;
	return 0;
}

void reduce(sir a, int n, sir b, int m, bool& posibil,int&k,int& nrMax)
{
	int lung_secv;
	k=0;
	nrMax=0;
	int ia=1,ib=1;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
	posibil=true;
	while((ia<=n)&&(posibil)&&(ib<=m)){
		lung_secv=determinaSecv(a, n, ia, b[ib]);
		if (lung_secv==0)
			posibil=false;
		else{
			if (lung_secv>nrMax){
				nrMax=lung_secv;
				k=ib;
			}
			ia=ia+lung_secv;
			ib++;
		}
	}
	if (posibil &&((ia<=n)||(ib<=m)) posibil=false;
}

int main(){
	sir a,b;	int n,m,k,nrMax;	bool posibil;
	cout<<"Introduceti sirul a"<<endl;
	citireSir(a,n);
	cout<<"Introduceti sirul b"<<endl;
	citireSir(b,m);
	reduce(a,n,b,m,posibil,k,nrMax);
	cout<<"Posibil ="<<posibil<<endl;
	if (posibil){
		cout<<"K= "<<k<<endl;
		cout<<"nrMax="<<nrMax<<endl;
	}
}
```

Implementare

Varianta Pascal

```
Program SecventeMaximePrimeIntreEle;
type vector=array[1..100] of integer;
procedure citireSir(var n:integer; var x:vector);
var i:integer;
begin
  writeln('Introduceti nr. de elemente');
  readln(n);
  writeln('Introduceti elementele');
  for i:=1 to n do
    begin
      read(x[i]);
```

23 noiembrie 2019 Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
    end;
end;

Function DeterminaLungSuma(n:integer;x:vector;poz, val:integer):integer;
var l,suma:integer;
begin
 l:=0; suma:=0;
 while (poz+l<=n) and (suma<val) do
    begin
     suma:=suma+x[poz+l];
     inc(l);
    end;
 if (suma=val) then
   DeterminaLungSuma:=l
 else
   DeterminaLungSuma:=0;
end;
Procedure Reduce(n:integer;a:vector;m:integer;b:vector; var posibil:boolean;var k,nrMax:integer);
var ia,ib,lung_secv:integer;
begin
          k:=0; nrMax:=0;
          ia:=1; ib:=1;
          posibil:=true;
          while (ia<=n) and (ib<=m) and posibil  do
          begin
                    lung_secv:=DeterminaLungSuma(n,a,ia,b[ib]);
                    if (lung_secv=0) then
                              begin
                              posibil:=false;
                             end
                    else
                             begin
                               if (lung_secv>nrMax) then
                                    begin
                                      nrMax:=lung_secv;
                                      k:=ib;
                                    end;
                                  ia:=ia+lung_secv;
                                   inc(ib);
                                  end;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
            end;

            if (posibil) and ((ia<=n) or (ib<=m)) then posibil:=false;
end;

Procedure TiparireRezultat(posibil:boolean; k,nrMax:integer);
var i,j:integer;
begin
  if (posibil) then
      writeln('Sirurile se pot reduce k=',k,' nrMax=',nrMax)
   else
      writeln('Sirurile nu se pot reduce');
end;
var n,m,k,nrMax:integer;
    a,b:vector;
    posibil:boolean;
begin
  citireSir(n,a);
  citireSir(m,b);
  Reduce(n,a,m,b,posibil,k,nrMax);
  TiparireRezultat(posibil,k, nrMax);
end.
```

Problema 5:

Se dă o mulţime de maxim 60 de numere naturale.

Se cer toate submulţimile disjuncte de numere *prietene* având cel puțin 2 elemente.

Spunem că două numere naturale *p* şi *q* sunt *prietene* dacă suma tuturor divizorilor lui *p* este egală cu suma tuturor divizorilor lui *q*.

Exemplu
M={68, 82, 64, 93, 127, 86, 131, 121, 137, 76, 139, 66, 70, 94, 115, 119, 149, 111, 151, 99, 125, 157, 133, 106, 163, 60, 78, 92, 123, 143, 167, 98, 173, 129, 88, 118, 145, 179, 117, 181, 169, 80, 122, 105, 141, 155, 161, 191,193, 57 }

Rezultat:
{ 68, 82 }
{ 93, 127 }
{ 86, 131 }

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

{ 76, 139 }
{ 66, 70, 94, 115, 119 }
{ 111, 151 }
{ 99, 125 }
{ 60, 78, 92, 123, 143, 167 }
{ 88, 118, 145, 179 }
{ 117, 181 }
{ 80, 122 }
{ 105, 141, 155, 161, 191 }

Analiza

Se construieste vectorul sd, cu proprietatea ca sd[i] este suma divizorilor elementului i din multimea data. Problema se reduce la a determina

Specificarea funcțiilor

Subalgoritmul **citireSir(n,x)**
Descriere: citeste sirul x cu n elemente
Date: -
Rezultate: n = numarul de elemente, x-elementele sirului

Subalgoritmul **tiparireSir(n,x)**
Descriere: timareste sirul x cu n elemente
Date: n = numarul de elemente, x-elementele sirului
Rezultate: -

Subalgoritmul **sumaDivizori**(x)
Descriere: Calculează suma divizorilor lui x
Date: x
Rezultate: suma div. lui x

Subalgoritmul **determinaPozitie**(n, x, val)
Descriere: determină poziţia unei valori in sirul x cu n elemente (daca exista, altfel returneaza -1)
Date: sirul x cu n elemente si o valoare cautata
Rezultate: Prima aparitie a val. in sirul x sau -1.

Subalgoritmul **creazaSirSd**(n, x, sd){
Descriere: Determină sumele div. corespunzatoare elementelor din x
Date: sirul x cu n elemente
Rezultate: sirul sumelor corespunzatoare

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Subalgoritmul **determinaMultimi**(n, sir x)
Descriere: construieste si tipareste submultimile de numere prietene
Date: sirul x cu n elemente
Rezultate: -

Implementare

Varianta C++

```
#include<iostream>
using namespace std;
#define MAX 60
typedef int sir[MAX];
void citesteSir(int &n, sir x){
        cout<<"Introduceti nr. de elemente ";
        cin>>n;
        cout<<"Introduceti elementele "<<endl;
        for(int i=1;i<=n;i++)
                cin>>x[i];
}
void tiparireSir(int n, sir x){
    for(int i=1;i<=n;i++)
        cout<<x[i]<<' ';
    cout<<endl;
}

int sumaDivizori(int x){
        int suma=1;
        for(int i=2;i<=x/2;i++)
                if (x %i==0)
                        suma+=i;
        if (x>1)
                suma+=x;
        return suma;
}

int determinaPozitie(int n, sir x, int val){
        for(int i=1;i<=n;i++)
                if (val==x[i])
                        return i;
        return -1;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
}
void creazaSirSd(int n, sir x, sir sd){
        for(int i=1;i<=n;i++)
                sd[i]=sumaDivizori(x[i]);
}

void determinaMultimi(int n, sir x){
        sir sd, determinate, multime;
        creazaSirSd(n,x,sd);
        int m=0;
        for(int i=1;i<n;i++){
                int lm=0;
                if (determinaPozitie(m,determinate,sd[i])<0){
                        determinate[++m]=sd[i];
                        multime[++lm]=x[i];
                        for(int j=i+1;j<=n;j++)
                                if (sd[i]==sd[j]){
                                        multime[++lm]=x[j];
                                }
                        if (lm>1)
                                tiparireSir(lm,multime);
                }

        }
}

 int main() {
 citesteSir(n,x);
 determinaMultimi(n,x);
}
```

Implementare

Varianta Pascal

```
Program SubmultimiNrPrietene;
type vector=array[1..100] of word;

procedure citireMultime(var n:integer; var x:vector);
var i:integer;
begin
  writeln('Introduceti nr. de elemente');    readln(n);
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
  writeln('Introduceti elementele');
  for i:=1 to n do   read(x[i]);
 end;

procedure tiparireMultime(n:integer;x:vector);
var i:integer;
begin
if n=0 then   writeln('Multimea vida')   else
  for i:=1 to n do    write(x[i],' ');
   writeln;
end;

function SumaDivizori(n:word):integer;
var suma,i:word;
begin
 suma:=1;
 for i:=2 to n div 2 do
   if (n mod i=0) then suma:=suma+i;
 if (n>1) then suma:=suma+n;
 SumaDivizori:=suma;
end;

function DeterminaPozitie(n:integer; x:vector;val:word):integer;
var i,poz:integer;
begin
  poz:=-1;    i:=1;
  while (i<=n) and (poz<0) do
     begin
        if (x[i]=val) then      poz:=i;
        inc(i);
      end;
   DeterminaPozitie:=poz;
end;

Procedure CreazaSirSd(n:integer; x:vector;var sd:vector);
var i:integer;
begin
 for i:=1 to n do
    sd[i]:=SumaDivizori(x[i]);
end;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
Procedure DeterminaMultimi(n:integer;x:vector);
var sd, determinate, multime:vector;
    i,j,lung_multime,lung_determinate:integer;
begin
CreazaSirSd(n,x,sd);
lung_determinate:=0;
for i:=1 to n-1 do
  begin
    lung_multime:=0;
    if (DeterminaPozitie(lung_determinate,determinate,sd[i])<0) then
       begin
         inc(lung_determinate);
         determinate[lung_determinate]:=sd[i];
         inc(lung_multime);
         multime[lung_multime]:=x[i];
         for j:=i+1 to n do
             if sd[i]=sd[j] then
               begin
                 inc(lung_multime);
                 multime[lung_multime]:=x[j]
               end;
         if (lung_multime>1) then tiparireMultime(lung_multime, multime);
       end;
  end;
end;

var n:integer;
   x:vector;
begin
   citireMultime(n,x);
   tiparireMultime(n,x);
   determinaMultimi(n,x);
end.
```

23 noiembrie 2019 Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Probleme tip grilă ...

1.Se consideră subalgoritmul h(n), unde ***n*** este un număr natural ($1 \le n \le 10000$)

```
Subalgoritmul h(A, n):
   Dacă n=0 atunci  returnează 0;
   SfDacă
   Dacă n mod 2=0 atunci returnează h(A,n-1)+A[n];
   SfDacă
   returnează  h(A,n-1)-A[n];
SfSubalgoritm
```

Algoritmul calculează:

a. Numărul elementelor pare din vectorul A
b. Suma elementelor de pe pozițiile pare din vectorul A
c. Diferența elementelor de pe pozițiile impare din vectorul A
d. Diferența dintre suma elementelor pare din vector și suma elementelor impare din vectorul A
e. Diferența dintre suma elementelor de pe poziții pare și suma elementelor de pe pozițiile impare din vectorul A
f. Niciunul din răspunsuri nu este corect

2. Fie șirul x=(5, 3, 2, 1, 1, 1). Ce va realiza următorul algoritm?

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar

Lect. Dr. Vasile Prejmarean

{Pascal}

```
for i:=1 to n do
begin
      c:=x[i]
      x[i]:= x[n-i+1];
      x[n-i+1]:=c;
end;
for i:=1 to n do
      Write (x[i],',');
```

//C

```
for (i=1;i<=n;i++)
{
      c=x[i];
      x[i]=x[n-i+1];
      x[n-i+1]=c;
}
for (i=1;i<=n;i++)
      printf("%d,", x[i]);
```

1. 1,1,2,1,3,5
2. 1,1,1,2,3,5
3. 5,3,2,1,1,1
4. Nici una dintre variantele anterioare nu este corecta.

3. Se consideră subalgoritmul f(n), unde ***n*** este un număr natural ($1 \le n \le 1000$) si $A=(A_0, A_1, A_2, \ldots, A_n)$ un șir de numere întregi:

```
Subalgoritmul f(A, n, q, i):
   Dacă n=i atunci
        returnează A[n];
```

23 noiembrie 2019 Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
    altfel
        returnează  f(A,n,q,i+1)*q+A[i];
    SfDaca
SfSubalgoritm
```

Dacă n=3, A=(0,-6,1,1) și se apelează subalgoritmul f(A,n,q,0) pentru ce valori ale lui q, subalgoritmul returnează valoarea 0?:

a. {0, 2, -3}
b. {0, 1, -2, 2}
c. {3, 1, -1}
d. {4, 0, 1, 5}

4. Se consideră subalgoritmul f(n), unde ***n*** este un număr natural ($1 \le n \le 1000$) si $A=(A_0, A_1, A_2, \ldots, A_n)$ un șir de numere întregi:

```
Subalgoritmul f(A, n, q, i):
    Dacă i=0 atunci
        returnează A[n-i];
    altfel
        returnează  f(A,n,q,i-1)*q+A[n-i];
    SfDaca
SfSubalgoritm
```

Dacă n=3, A=(0,-6,1,1) și se apelează subalgoritmul f(A,n,q,n) pentru ce valori ale lui q, subalgoritmul returnează valoarea 0?:

a. {0, 1, -2, 2}

23 noiembrie 2019 Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

b. {3, 1, -1}
c. {4, 0, 1, 5}
d.{0, 2, -3}

5. *Media aritmetică a elementelor **mofturoase** la divizori, dar nu la 2!* Care dintre următoarele patru funcţii calculează **corect** *media aritmetică* a elementelor "***mofturoase***" dintr-un şir *X* cu *n* elemente numere naturale strict pozitive, care are cel puţin un element "***mofturos***" ? Spunem că un număr natural este "***mofturoase***" dacă este *Putere a lui doi* (se poate scrie sub forma $\mathbf{2^p}$, ***p*** fiind un număr natural ≥ 0, singurul divizor prim permis este **2**!).

   Exemple:

   - Pentru şirul *X* = (3, **8**, 10, **16**, 13, **2**, **1**), *n* =7,
     se va calcula *media aritmetică* a elementelor **8, 16, 2, 1** = **6.75**.
   - Pentru şirul *X* = (5, **8**, 10, **16**, 31, **32**, **1024**, 32769, 1048575, **64**, 13, **2**, **1**), *n* =13,
     se va calcula *media aritmetică* a elementelor 8, 16, 32, 1024, 64, 2, 1 = **163.857**.

Raspuns:

a) `Ma_a` ? *Nu !*
b) `Ma_b` ? *Da !*
c) `Ma_c` ? *Da !*
d) `Ma_a` ? *Da !*

Obs. Cel putin trei variante sunt corecte si cel putin una este gresita !

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar

Lect. Dr. Vasile Prejmarean

Varianta C++

```
//   *  Var. a  *   \\

int    f(int x)  { return x? f(x/2)+x%2:0; }

bool   a(int x)  { return f(x)==1; }

double Ma_a(int* X, int n)
{
      int S2p=0, Nr2p=0;
      for (int i=1; i<=n; i++)
           if (a(X[i]))  S2p+=X[i],
                         Nr2p++;
      return S2p/Nr2p;                        //  *(d_d)*
}
```

```
//   *  Var. b  *   \\

bool   b(int x) { return x==1 or (b(x/2) and not(x%2)); }
void   X_Y(int* X, int k, int* Y, int& m)
{
       if (k) { X_Y(X,k-1,Y,m);
                if (b(X[k]))  Y[++m]=X[k]; }  else m=0;
}
double MaR(int* X, int n)
{
      return n? (MaR(X,n-1)*(n-1)+X[n])/n: 0;
}
double Ma_b(int* X, int n)
{
      int  Y[n+1], Nr2p;
         X_Y(X,n,Y,Nr2p);
      return MaR(Y,Nr2p);
}
```

```
//   *  Var. c  *   \\

bool   c(int x)  { return x<2 or (c(x/2) and x%2-1); }

double Ma_C(int* X, int n, int& k)
{
       if (n) { double Ma_k_1=Ma_C(X,n-1,k);
                return c(X[n])? (Ma_k_1*k+X[n])/++k:
                                Ma_k_1;             }
       else  return 123456789;
}

double Ma_c(int* X, int n, int k=0)
{
       return Ma_C(X,n,k);
}
```

```
//   *  Var. d  *   \\

bool   d(int x, int y=2)  { return x<=2 or x==y or
                                  (x>y and d(x,y*2)); }

double Ma_D(int* X, int n, int& k)
{
       return n? c(X[n])? (Ma_D(X,n-1,k)*k+X[n])/++k:
                          Ma_D(X,n-1,k):  987654321;
}


double Ma_d(int* X, int n)
{                    int k;
       return Ma_D (X,n,k=0);
}
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar

Lect. Dr. Vasile Prejmarean

Varianta Pascal

```
//   *  Var. a  *   \\
Function f(x:Integer):Integer;
Begin
        if (x>0) Then f:=f(x Div 2) + x Mod 2
                 Else f:=0;
End;

Function a(x:Integer):Boolean;
Begin
        a:=f(x)=1;
End;

Type     Sir = Array[1..100] Of Integer;

Function Ma_a(X:Sir; n:Integer):Real;
Var   S2p, Nr2p, i:Integer;
Begin
      S2p:=0; Nr2p:=0;
      For i:=1 To n Do
          If a(X[i])  Then Begin S2p:=S2p+X[i];
                                 Nr2p:=Nr2p+1 End;
      Ma_a:=S2p/Nr2p;                                  // *(-_-)* Ew
End;
```

```
//   *  Var. b  *   \\
Function b(x:Integer):Boolean;
Begin
        b:= (x=1) or (b(x div 2) and (x mod 2=0))
End;
Procedure X_Y(X:Sir; k:Integer; Var Y:Sir; Var m:Integer);
Begin
      if k>0 Then Begin X_Y(X,k-1,Y,m);
                        If b(X[k]) Then Begin m:=m+1;
                                             Y[m]:=X[k]
                   End                  End
              Else m:=0;
End;
Function MaR(X:Sir; n:Integer):Real;
Begin
        if n>0 then MaR:=(MaR(X,n-1)*(n-1)+X[n])/n
               else MaR:=0
End;
Function Ma_b(X:Sir; n:Integer):Real;
Var     Y:Sir; Nr2p:Integer;
Begin
        X_Y(X,n,Y,Nr2p);    Ma_b:=MaR(Y,Nr2p)
End;
```

```
//   *  Var. c  *   \\
Function c(x:Integer):Boolean;
Begin
        c:=(x<2) or (c(x Div 2) and (x Mod 2 = 0) )
End;
Function Ma_Cc(X:Sir; n:Integer; Var k:Integer):Real;
Var     Ma_k_1:Real;
Begin
       if n>0 Then Begin Ma_k_1:=Ma_Cc(X,n-1,k);
              If c(X[n]) Then Begin Ma_Cc:=(Ma_k_1*k+X[n])/(k+1);
                                    k:=k+1 End
                         Else     Ma_Cc:= Ma_k_1;
                 End
              Else  Ma_Cc:= 8989;
End;
Function Ma_c(X:Sir; n,k:Integer):Real;
```

```
//   *  Var. d  *   \\
Function d(x, y:Integer):Boolean;
Begin
        d:=(x<=2) or (x=y) or ((x>y) and d(x,y*2))
End;
Function Ma_D(X:Sir; n:Integer; Var k:Integer):Real;
Begin
       if n>0 then if d(X[n],2) then Begin
                     Ma_D:=(Ma_D(X,n-1,k)*k+X[n])/(k+1);
                      k:=k+1          End
                                  else Ma_D:= Ma_D(X,n-1,k)
              else Ma_D:=8990
End;
Function Ma_d(X:Sir; n:Integer):Real;
Var    k:Integer;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
Begin
        Ma_c:= Ma_Cc(X,n,k);
End;
```

```
Begin
      k:=0;    Ma_d:=Ma_D(X,n,k)
End;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar

Lect. Dr. Vasile Prejmarean

6. Care dintre următoarele patru funcţii determină numărul format din primele *k* cifre care îndeplinesc o anumită proprietate (dată printr-o funcţie booleană *Pr(c)*), plecând de la (pentru) un număr natural *n* (dat).

   Exemple:

   - Pentru *n* = 3 . **8** 5 **4** . 1 **6 0** , *k* =3 şi proprietatea *Pr(c)* : *c* este o cifră pară, rezultatul este **846,**
   - Pentru *n* = 3 . **8** 5 7 . 1 **6** 3 , *k* =3 şi proprietatea *Pr(c)* : *c* este o cifră pară, rezultatul este **86**,
   - Pentru *n* = 3 . 5 5 7 . 1 1 3 , *k* =3 şi proprietatea *Pr(c)* : *c* este o cifră pară, rezultatul este **0**,

Raspuns:

a) __ ? *Da !* **f_a**
b) __ ? *Da !* **f_b**
c) __ ? *Da !* **f_c**
d) __ ? *Da !* **f_d**

Obs. Cel putin patru variante sunt corecte si cel mult una este gresita !

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

Varianta C++

```
#include <iostream>          ///    n=1296385240, k=3;  -> 268
using namespace std;
bool Pr(int c)  { return c%2==0; }
```

```
//   *  Var. a  *   \\

int  f_a(int n, int k, int& m)
{
     if (n) {   int r = f_a(n/10,k,m);
                return m<k and Pr(n%10)? r*10+n%10+m/k*m++: r;
            }
         else   return 0;
}
```

```
//   *  Var. b  *   \\

int  Last(int n, int k)
{
     int m=0;
     while (n and k) {
            if (Pr(n%10)) m=m*10+n%10, k--;
            n/=10;   }
     return m;
}
int  Inv(int x, int y=0)
{
     return x? Inv(x/10,y*10+x%10) :y;
}
int  f_b(int n, int k)
{
     return Last(Inv(n*10+1),k);
}
```

```
//   *  Var. c  *   \\

int  Last(int n, int k, int& m)
{
     return n and k? Pr(n%10)? Last(n/10,k-1,m=m*10+n%10):
                               Last(n/10,k,m): m;
}
int  f_c(int n, int k, int& m )
{
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
      return Last(Inv(n*10+1),k,m);
}
```

```
//    *  Var. d  *    \\

int  All(int n)
{
      return n? Pr(n%10)? All(n/10)*10+n%10: All(n/10) :0;
}

int  Log(int n)
{
      return n? Log(n/10)+1 :0;
}

int  Tai(int n, int k)
{
      return k? Tai(n/10,k-1): n;
}

int  f_d(int n, int k)
{
      int a=All(n);   return Tai(a,max(Log(a)-k,0));
}
```

```
void Print(int n, int k, int r)
{
      cout << "\n Pentru n = " << n << " si k = " << k
           << " Nr.(n,k) = "   << r << endl;
}

void Var_a(int n, int k, int m=0)   {   Print(n,k,f_a(n,k,m)); }
void Var_b(int n, int k)            {   Print(n,k,f_b(n,k  )); }
void Var_c(int n, int k, int m=0)   {   Print(n,k,f_c(n,k,m)); }
void Var_d(int n, int k)            {   Print(n,k,f_d(n,k  )); }

int main()
{
  int n[] = { 5, 3854160,3857163,3557113,53071092,29685240 }, k=3;

  for (int i=1; i<=n[0]; i++)
       Var_a(n[i],k),
       Var_b(n[i],k),
       Var_c(n[i],k),
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
        Var_d(n[i],k),          cout << endl;
}
```

Varianta Pascal

```
Program  Pr_c_;                          //    n=12638, k=3;  -> 268

Function Pr(c:Integer):Boolean;          //   Pr.:  c este par
Begin
         Pr:=(c Mod 2) = 0
End;
```

```
                        //   *  Var. a  *   \\

Function f_a(n, k:Integer; Var m:Integer):Integer;
Var r:Integer;
Begin
     If n>0 Then Begin r:= f_a(n Div 10,k,m);
                    If (m<k) and Pr(n Mod 10)
                       Then Begin
                              f_a:=r*10+n Mod 10+m Div k*m;
                              Inc(m)
                            End
                       Else f_a:=r;
                 End
            Else   f_a:=0
End;
```

```
                        //   *  Var. b  *   \\

Function Lastb(n, k:Integer):Integer;
Var  m:Integer;
Begin
     m:=0;
     While ((n>0) and (k>0)) Do Begin
           If Pr(n Mod 10) Then Begin m:=m*10+(n Mod 10);
                                      k:=k-1 End;
           n:=n Div 10        End;
     Lastb:=m
End;

Function Inv(x,y:Integer):Integer;
Begin
      If x>0 Then Inv:=Inv(x Div 10,y*10+x Mod 10) Else Inv:=y;
End;

Function f_b(n, k:Integer):Integer;
Begin
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
          f_b:=Lastb(Inv(n*10+1,0),k)
End;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
                        //   *  Var. c  *   \\

Function Last(n, k: Integer; Var m: Integer): Integer;
Begin
        If (n>0) and (k>0)
            Then
                 If Pr(n Mod 10) Then           Begin
                         m:=m*10+n Mod 10;
                         Last:=Last(n Div 10,k-1,m) End
                                 Else  Last:=Last(n Div 10,k,m)
            Else Last:=m
End;

Function f_c(n, k: Integer; Var m: Integer): Integer;
Begin
         f_c:=Last(Inv(n*10+1,0),k,m);
End;
```

```
                        //   *  Var. d  *   \\

Function All(n:Integer):Integer;
Begin
      If n>0 Then
         If Pr(n Mod 10) Then All:=All(n Div 10)*10+n Mod 10
                         Else All:=All(n Div 10)
             Else All:=0;
End;

Function Log(n:Integer):Integer;
Begin
      If n>0 Then Log:=Log(n Div 10)+1 Else Log:=0
End;

Function Tai(n,k:Integer):Integer;
Begin
      If k>0 Then Tai:=Tai(n Div 10,k-1) Else Tai:=n;
End;

Function Max(a,b:Integer):Integer;
Begin
         If a>b Then Max:=a Else Max:=b
End;

Function f_d(n,k:Integer):Integer;
Var  a:Integer;
Begin
     a:=All(n);  f_d:=Tai(a,Max(Log(a)-k,0))
End;
```

23 noiembrie 2019

Conf. Dr. Grigoreta Cojocar
Lect. Dr. Vasile Prejmarean

```
Procedure Print(n,k,r:Integer);
Begin
     Writeln(' Pentru n = ',n,' si k = ',k,' Nr.(n,k) = ',r)
End;

Procedure Var_a(n,k,m:Integer); Begin Print(n,k,f_a(n,k,m)) End;
Procedure Var_b(n,k  :Integer); Begin Print(n,k,f_b(n,k  )) End;
Procedure Var_c(n,k,m:Integer); Begin Print(n,k,f_c(n,k,m)) End;
Procedure Var_d(n,k  :Integer); Begin Print(n,k,f_d(n,k  )) End;

Const m=5; n:Array[1..m] Of Integer = (1846, 1863, 113,3002,1268); k=3;
Var   i:Integer;
Begin
      For i:=1 To m Do Begin

          Var_a(n[i],k,0); Var_b(n[i],k);

          Var_c(n[i],k,0); Var_d(n[i],k);  Writeln

      End;

      Readln
End.
```

Rezultatele (pentru toate variantele) vor fi următoarele:

```
Pentru n = 1846 si k = 3 Nr.(n,k) = 846
Pentru n = 1863 si k = 3 Nr.(n,k) =  86
Pentru n =  113 si k = 3 Nr.(n,k) =   0
Pentru n = 3002 si k = 3 Nr.(n,k) =   2
Pentru n = 1268 si k = 3 Nr.(n,k) = 268
```