



Limbaje de programare I

Culegere de probleme ---- M.Marcu

C/C++

Limbaje de programare I

Laborator 1 – Introducere in C

Probleme rezolvate:

a) Afisarea mesajului “Hello world!” pe ecran.

```
#include <stdio.h>

//directiva preprocesor de includere a fisierului corespunzator

int main()
{
/* comentariu */
printf("Hello, World! \n");
return 0;

//functia main principala returneaza valoarea 0
}
```

b) Rezolvarea ecuatiei de gradul II cu coeficienti reali, care se citesc de la tastatura.

```
#include <stdio.h>
#include <stdio.h>
#include <math.h>
int main ()
{
/* variabila locala */
float a=0;
float b=0;
float c=0;
float x1=0,x2=0;
float delta=0,delt=0;
```

```
printf("\nEcuatie de gradul doi ax^2+bx+c=0\n");
printf("\na:=");
scanf("%f",&a);
printf("\nb:=");
scanf("%f",&b);
printf("\nc:=");
scanf("%f",&c);
delt=b*b-4*a*c;
printf("\nDelta:=%f",delt);
if(delt<0)
{
delta=sqrt(-delt);
printf("\nSolutiile ecuatiei sunt x1,2=%f+-i%f",-b/(2*a),delta/(2*a));
}
if(delt>0)
{
delta=sqrt(delt);
x1=(-b+delta)/(2*a);
x2=(-b-delta)/(2*a);
printf("\nSolutiile ecuatiei sunt x1,2=%f,%f",x1,x2);
}
if(delt==0)
{
delta=sqrt(delt);
x1=(-b+delta)/(2*a);
printf("\nSolutiile ecuatiei sunt x1,2=%f",x1);
}
```

```
getche();  
return 0;  
}
```

c) Scrieți un program care să determine cadranul în care se află un punct dat prin coordonatele sale (x,y) (întregi sau reale)

```
#include <stdio.h>  
  
int main(void)  
{  
    int x,y;  
    printf("\n x:=");  
    scanf("%d", &x);  
    printf("\n y:=");  
    scanf("%d", &y);  
    if(x>=0 && y>0)  
        printf("Primul cadran");  
    if(x>=0 && y<=0)  
        printf("Al doilea cadran");  
    if(x<0 && y>=0)  
        printf("Al treilea cadran");  
    if(x<=0 && y<=0)  
        printf("Al patrulea cadran");  
    return 0;  
}
```

Probleme propuse:

- 1) Scrieți un program care să verifice dacă 3 numere întregi date a,b,c sunt într-o progresie aritmetica sau geometrica.
- 2) Scrieți un program care determină minimul și maximul dintre 4 numere citite de la tastatura, folosind doar instrucțiunea if.
- 3) Scrieți un program care calculează dacă două numere citite de la tastatura sunt divizibile între ele.

Limbaje de programare |

Laborator 2 - Instructiuni si expresii conditionale

Probleme rezolvate:

a) Se citeste de la tastatura un numar intreg. Determinati daca este prim sau nu. In cazul in care numarul nu este prim, afisati divizorii acestuia.

```
#include <stdio.h>

int main(void)
{
    int nr, i;
    int prim = 1; //prim este 1

    printf("\n Nr:=");
    scanf("%d", &nr);

    for (i = 2; i < nr; i++) {
        if (nr % i == 0) {
            printf("Divizorul este:=%d \n",i);
            prim = 0; //are divizori
        }
    }

    if (nr > 1 && prim) {
        printf("Numarul %d este numar prim", nr);
    }
    else{
        printf("Numarul %d nu este numar prim!",nr);
    }
}
```

```
printf("\n");  
return 0;  
}
```

Probleme propuse:

- 1) Modificati programul anterior astfel incat sa afisati toate numerele prime dintr-un interval [A,B].
- 2) Se citeste un numar N de la tastatura, iar mai apoi N numere intregi. Sa se afiseze suma tuturor numerelor prime.
- 3) Modificati programul anterior astfel incat sa afisati suma si produsul divizorilor unui numar citit de la tastatura.

Limbaje de programare

Laborator 3 - Tablouri unidimensionale - vectori

Probleme rezolvate:

a) Se citeste de la tastatura un numar intreg N, iar mai apoi se citesc N numere intregi intr-un vector. Se cere sa se afiseze continutul tabloului, precum si minimul, respectiv maximul din tabloul unidimensional.

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int n, i, minim=99999, maxim=-99999;
    int v[100]; // vectorul are maxim 100 de intregi

    printf("Nr de elemente:=\n");
    scanf("%d", &n); // citeste nr de elemente

    for (i = 0; i < n; i++) {
        printf("\n v[%d]:=", i);
        scanf("%d", &v[i]); // citire elemente
    }

    for (i = 0; i < n; i++) {
        printf("v[%d]=%d", i, v[i]); // scrie elemente
    }

    for(i=0;i<n;i++)
    {
```

```
    if (v[i]>maxim)
        { maxim=v[i];
        }
    if (v[i]<minim){
        minim=v[i];
    }
}
printf("\nMinimul este %d",minim);
printf("\nMaximul este %d",maxim);
return 0;
}
```

b) Modificati programul anterior astfel incat sa afisati numerele pare din vector

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int n, i,minim=99999,maxim=-99999;
    int v[100]; // vectorul a are maxim 100 de intregi

    printf("Nr de elemente:=\n");
    scanf("%d", &n); // citeste nr de elemente

    for (i = 0; i < n; i++) {
        printf("\n v[%d]:=",i);
        scanf("%d", &v[i]); // citire elemente
    }
}
```

```
}

for (i = 0; i < n; i++) {
    if (v[i]%2==0)printf("\nv[%d]=%d",i,v[i]); // scrie elemente
}

for(i=0;i<n;i++)
{
    if (v[i]>maxim)
        { maxim=v[i];
        }
    if (v[i]<minim){
        minim=v[i];
    }
}
printf("\nMinimul este %d",minim);
printf("\nMaximul este %d",maxim);
return 0;
}
```

Probleme propuse:

- 1) Modificati programul anterior astfel incat sa afisati suma numerelor pare si produsul numerelor impare.
- 2) Se citesc un numar N de numere intregi. Se cere sa se sorteze elementele vectorului, iar mai apoi folosind programul rezolvat, sa se determine daca aceste numere sunt intr-o progresie aritmetica.

Laborator 4 - Tipul tablou - matrice

Probleme rezolvate:

a) Se citesc de la tastatura doua numere m si n. Se cere sa se defineasca o matrice cu m linii si n coloane care sa fie citita de la tastatura, iar mai apoi afisata.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n,m,i,j,a[20][20];
    printf("\n m:=");
    scanf("%d",&m);
    printf("\n n:=");
    scanf("%d",&n);
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
            printf("\na[%d][%d]:=",i,j);
            scanf("%d",&a[i][j]);
        }
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
            printf("\na[%d][%d]:=%d",i,j,a[i][j]);
        }
}
```

```
    return 0;
}
```

b) Determinati minimul, respectiv maximul din aceasta matrice.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n,m,i,j,a[20][20];
    int minim=99999,maxim=-99999;
    printf("\n m:=");
    scanf("%d",&m);
    printf("\n n:=");
    scanf("%d",&n);
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
            printf("\na[%d][%d]:=",i,j);
            scanf("%d",&a[i][j]);
        }

    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
```

```
printf("\na[%d][%d]:=%d",i,j,a[i][j]);  
if(a[i][j]>maxim) maxim=a[i][j];  
if(a[i][j]<minim) minim=a[i][j];  
}  
printf("Minim:=%d, maaxim:=%d",minim,maxim);  
return 0;  
}
```

Probleme propuse:

- 1) Se citesc doua matrici A si B. Sa se afiseze suma si diferenta dintre cele doua matrice.
- 2) Se considera citirea a doua matrice patratice A si B. Sa se determine matricea C, care reprezinta produsul dintre cele doua matrice, $C = A \times B$.
- 3) Sa se calculeze suma elementelor de pe diagonala principala, respectiv produsul de pe diagonala secundara a unei matrice patratice A.

Laborator 5 - Functii

Probleme rezolvate:

a) Se citeste de la tastatura un sir de N numere reale. Folosind functii, determinati rezultatul expresiei ax^2+bx+c pentru fiecare numar din sir.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
float functie(float x, float a, float b, float c)
```

```
{
```

```
    return (a*x*x+b*x+c);
```

```
}
```

```
int main()
```

```
{
```

```
    int i,n;
```

```
    float vector[200],a,b,c;
```

```
    printf("\n n:=");
```

```
    scanf("%d",&n);
```

```
    printf("\n a:=");
```

```
    scanf("%f",&a);
```

```
    printf("\n b:=");
```

```
    scanf("%f",&b);
```

```
    printf("\n c:=");
```

```
    scanf("%f",&c);
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        printf("\n vector[%d]:=",i);
```

```
scanf("%f",&vector[i]);
}

for(i=0;i<n;i++)
{
printf("\n f[vector[%d]]:=%f",i,functie(vector[i],a,b,c));
}
return 0;
}
```

b) Folosind functii, sa se determine numarul de cifre pentru un numar citit de la tastatura.

```
#include <stdio.h>

int numarcifre(int numar)
{
int nr = 0;
while (numar)
{
numar =(int)numar/10;
nr++;
}
return nr;
}
```

```
int main(void)
```



```
{  
    int numar;  
    printf("Numarul:= ");  
    scanf("%d", &numar);  
    printf("Numarul %d are %d cifre!\n", numar, numarcifre(numar));  
    return 0;  
}
```

Probleme propuse:

- 1) Folosind functii, determinati cifra minima dintr-un numar citit de la tastatura.
- 2) Folosind functii, determinati cifra maxima dintr-un numar citit de la tastatura.
- 3) Scrieti o functie care determina cel mai mare divizor comun pentru doua numere citite de la tastatura.
- 4) Scrieti o functie care determina cel mai mic multiplu comun pentru doua numere citite de la tastatura.

Laborator 6 - Metode de sortare

a) Metoda sortarii selective

```
#include <iostream>
using namespace std;
void sortareselectiva(int vv[], int N)
{
    for(int i = 0; i < N - 1; i++)
    {
        for(int j = i + 1; j < N; j++)
        {
            if(vv[i] > vv[j])
            {
                int aux = vv[i];
                vv[i] = vv[j];
                vv[j] = aux;
            }
        }
    }
}
void afisare(int vv[], int N)
{
    for(int i = 0; i < N; i++)
        cout << vv[i] << " ";
    cout << "\n";
}
int main()
```

```
{
    int vec[100];
    int N;
    cout << "Lungimea vectorului este: ";
    cin >> N;
    for(int i = 0; i < N; i++)
    {
        cout << "Elementul vector[" << i << "] este: ";
        cin >> vec[i];
    }
    sortareselectiva(vec, N);
    afisare(vec, N);
    return 0;
}
```

b) Metoda bulelor

```
#include<iostream>
using namespace std;
int main()
{
    int n,i,aux,terminat,vecto[50];
    cout<<"Introduceti dimensiunea a a vectorului : ";
    cin>>n;
    for(i=0;i<=n-1;i++) {
        cout<<"vector["<<i<<"]="";
```

```
cin>>vecto[i];}
terminat=0;
while(!terminat)
{
terminat=1;
for(i=0;i<n-1;i++)
if(vecto[i]>vecto[i+1]) {
aux=vecto[i];
vecto[i]=vecto[i+1];
vecto[i+1]=aux;
terminat=0;}
}
cout<<"Vectorul ordonat este : ";
for(i=0;i<=n-1;i++)
cout<<vecto[i]<<" ";
cout<<endl;
return 0;
}
```

Probleme propuse:

- 1) Sa se realizeze sortarea crescatoare a elementelor vectorului aflate pe pozitile impare, respectiv sortarea descrescatoare e elementelor vectorului care se gasesc pe pozitile pare.
- 2) Sa se determine suma elementelor care in urma sortarii se afla pe pozitii pare, respectiv impare.
- 3) Sa se determine suma elementelor impare care in urma sortarii se afla pe pozitii corespunzatoare unor numere prime, respectiv ne-prime.

Laborator 7 Principiile programarii in C++ - clase

Se considera urmatoarul program care defineste o structura de clasa care contine un set de numere intregi. Programul citeste si afiseaza de la tastatura un set de numere intregi.

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

class Numere{
public:
    int vec[200];
    int nr;
    void citire();
    void afisare();
};

void Numere::citire() {
    cout << "n:=" << endl;
    cin >> this->nr;
    for(int i=0;i<nr;i++)
    {
        cout << "vec["<<i<<"]:=";
        cin >> this->vec[i];
    }
}

void Numere::afisare() {
    cout << "n:=" << this->nr << endl;
    for(int i=0;i<nr;i++)
```

```
{
    cout << "vec["<<i<<"]:=" << vec[i] << endl;
}
}
```

```
int main() {
    Numere obiect;
    obiect.citire();
    obiect.afisare();
    return 0;
}
```

b) Completati programul urmatoare cu o functie care sa sorteze elementele vectorului crescator/descrescator in functie de un parametru.

```
#include <iostream>
using namespace std;
class Numere{
public:
    int vec[200];
    int nr;
    void citire();
    void afisare();
    void sortare(int parametru);
};
```

```
void Numere::citire() {
    cout << "n:=" << endl;
```

```
cin >> this->nr;
for(int i=0;i<nr;i++)
{
    cout << "vec["<<i<<"]:=";
    cin >> this->vec[i];
}
}
```

```
void Numere::afisare() {
    cout << "n:=" << this->nr << endl;
    for(int i=0;i<nr;i++)
    {
        cout << "vec["<<i<<"]:= " << vec[i] << endl;
    }
}
```

```
void Numere::sortare(int parametru) {
    if(parametru==1)
    {
        for(int i=0;i<nr-1;i++)
        for(int j=i+1;j<nr;j++)
        if(this->vec[i]<this->vec[j])
        {int aux=this->vec[i];
        this->vec[i]=this->vec[j];
        this->vec[j]=aux;
        }
    }
}
```

```
else{
    for(int i=0;i<nr-1;i++)
    for(int j=i+1;j<nr;j++)
    if(this->vec[i]>this->vec[j])
        {int aux=this->vec[i];
        this->vec[i]=this->vec[j];
        this->vec[j]=aux;
        }
    }
}
```

```
int main() {
    Numere obiect;
    obiect.citire();
    obiect.afisare();
    obiect.sortare(1);
    obiect.afisare();
    obiect.sortare(0);
    obiect.afisare();
    return 0;
}
```

Probleme propuse:

- 1) Sa se completeze structura de clasa cu doua functii care determina numarul de numere pare, respectiv impare din clasa.
- 2) Sa se completeze structura de clasa cu o functie care sorteaza elementele vectorului folosind pointeri.

3) Sa se completeze structura de clasa cu o functie care sorteaza elementele vectorului din laboratorul precedent.

Limbaje de programare I

Laborator 8 - Constructori

Se considera urmatoarul program care defineste o structura de clasa care contine un set de numere intregi. Programul citeste si afiseaza de la tastatura un set de numere intregi.

```
#include <iostream>

using namespace std;

class Numere{
public:
    int *vec;
    int *nr=new int;
    Numere();//definire constructor
    void citire();
    void afisare();
};

void Numere::citire() {
    cout << "n:=" << endl;
    cin >> *nr;
    vec=new int[200];
    for(int i=0;i<*nr;i++)
    {
        cout << "vec["<<i<<"]:=";
        cin >> vec[i];
    }
}
```

```
void Numere::afisare() {  
    cout << "n:=" << *nr << endl;  
    for(int i=0;i<*nr;i++)  
    {  
        cout << "vec["<<i<<"]:=" << vec[i] << endl;  
    }  
}
```

```
Numere::Numere() {  
}
```

```
int main() {  
    Numere obiect;  
    obiect.citire();  
    obiect.afisare();  
    return 0;  
}
```

Probleme rezolvate:

a) Completati constructorul cu un mesaj de afisare.

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

```
class Numere{
public:
    int *vec;
    int *nr=new int;
    Numere();//definire constructor
    void citire();
    void afisare();
};

void Numere::citire() {
    cout << "n:=" << endl;
    cin >> *nr;
    vec=new int[200];
    for(int i=0;i<*nr;i++)
    {
        cout << "vec["<<i<<"]="";
        cin >> vec[i];
    }
}

void Numere::afisare() {
    cout << "n:=" << *nr << endl;
    for(int i=0;i<*nr;i++)
    {
        cout << "vec["<<i<<"]=" << vec[i] << endl;
    }
}
```

```
}  
Numere::Numere() {  
    cout << "Initializare obiect!!" << endl;  
}  
int main() {  
    Numere obiect;  
    obiect.citire();  
    obiect.afisare();  
    return 0;  
}
```

b) Realizati citirea de la tastatura a tuturor numerelor in constructorul clasei.

```
#include <iostream>  
using namespace std;  
  
class Numere{  
public:  
    int *vec;  
    int *nr;  
    Numere();//definire constructor  
    void citire();  
    void afisare();  
};
```

```
void Numere::citire() {  
}
```

```
void Numere::afisare() {  
    cout << "n:=" << *nr << endl;  
    for(int i=0;i<*nr;i++)  
    {  
        cout << "vec["<<i<<"]:=" << vec[i] << endl;  
    }  
}
```

```
Numere::Numere() {  
    cout << "Initializare obiect!!" << endl;  
    cout << "n:=" << endl;  
    nr=new int;  
    cin >> *nr;  
    vec=new int[200];  
    for(int i=0;i<*nr;i++)  
    {  
        cout << "vec["<<i<<"]:=";  
        cin >> vec[i];  
    }  
}
```

```
int main() {
```

```
Numere obiect;  
obiect.afisare();  
return 0;  
}
```

Probleme propuse:

- 1) Realizati instantierea unui al doilea obiect de tip Numere, citit de la tastatura si mai apoi afisat.
- 2) Folosind operatorul delete (pentru vectori delete[] nume_variabila) realizati o functie care sterge toata colectia de date din aceasta clasa.
- 3) Modificati programul astfel incat clasa sa contina inca un sir de numere intregi, sir care va fi citit de la tastatura in constructorul clasei.

Limbaje de programare |

Laborator 9 - Destructori

Se considera urmatoarul program care defineste o structura de clasa care contine un set de numere intregi. Programul citeste de la tastatura si afiseaza un set de numere intregi.

```
#include <iostream>

using namespace std;

class Numere{

public:

    int *vec;

    int *nr;

    Numere();//definire constructor

    ~Numere();//definire destructor

    void citire();

    void afisare();

};

void Numere::citire() {

}

void Numere::afisare() {

    cout << "n:=" << *nr << endl;

    for(int i=0;i<*nr;i++)

    {

        cout << "vec["<<i<<"]:= " << vec[i] << endl;

    }

}
```



```
}  
Numere::Numere() {  
    cout << "Initializare obiect!!" << endl;  
    cout << "n:=" << endl;  
    nr=new int;  
    cin >> *nr;  
    vec=new int[200];  
    for(int i=0;i<*nr;i++)  
    {  
        cout << "vec["<<i<<"]=";  
        cin >> vec[i];  
    }  
}  
  
int main() {  
    Numere obiect;  
    obiect.afisare();  
    return 0;  
}
```

Probleme rezolvate:

a) Completati clasa cu destructorul acesteia, in care se va scrie doar un mesaj de afisare.

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

```
class Numere{
public:
    int *vec;
    int *nr;
    Numere();//definire constructor
    ~Numere();
    void citire();
    void afisare();
};

void Numere::citire() {
}

void Numere::afisare() {
    cout << "n:=" << *nr << endl;
    for(int i=0;i<*nr;i++)
    {
        cout << "vec["<<i<<"]:= " << vec[i] << endl;
    }
}

Numere::Numere() {
    cout << "Initializare obiect!!" << endl;
    cout << "n:=" << endl;
}
```

```
nr=new int;
cin >> *nr;
vec=new int[200];
for(int i=0;i<*nr;i++)
{
    cout << "vec["<<i<<"]=";
    cin >> vec[i];
}
}
```

```
Numere::~Numere() {
    cout << "Apelare destructor!!" << endl;
}
int main() {
    Numere obiect;
    obiect.afisare();
    obiect.~Numere();//apelare destructor implicit!
    return 0;
}
```

b) Completati clasa cu destructorul acesteia, in care se vor sterge toate variabilele de clasa.

```
#include <iostream>
using namespace std;
class Numere{
```

```
public:
    int *vec;
    int *nr;
    Numere();//definire constructor
    ~Numere();
    void citire();
    void afisare();
};

void Numere::citire() {
}

void Numere::afisare() {
    cout << "n:=" << *nr << endl;
    for(int i=0;i<*nr;i++)
    {
        cout << "vec["<<i<<"]= " << vec[i] << endl;
    }
}

Numere::Numere() {
    cout << "Initializare obiect!!" << endl;
    cout << "n:=" << endl;
    nr=new int;
    cin >> *nr;
    vec=new int[200];
}
```

```

for(int i=0;i<*nr;i++)
{
    cout << "vec["<<i<<"]=";
    cin >> vec[i];
}
}
Numere::~Numere() {
    cout << "Apelare destructor!!" << endl;
    delete this->nr;
    delete[] this->vec;
}
int main() {
    Numere obiect;
    obiect.afisare();
    obiect.~Numere();//apelare destructor implicit!
    return 0;
}

```

Probleme propuse

- 1) Adaugati o functie recursiva pentru calculul factorialului unui numar.
- 2) Pentru sirul de numere intregi, folosind functia de la punctul anterior, sa se retina intr-un nou vector de numere intregi factorialele corespunzatoare.
- 3) Completati destructorul clasei cu stergerea variabilelor suplimentare.
- 4) Realizati o functie recursiva in care se va determina numarul lui Fibonacci. Numerele Fibonacci sunt definite prin urmatoarea relatie de recurenta:

$$F\{0\}=0, F\{1\}=1, F\{i\}=F\{i-1\}+F\{i-2\}$$

Laborator 10 - Mostenire in C++

Se considera urmatorul program care implementeaza o relatie de mostenire intre doua clase.

```
#include <fstream>

#include <iostream>

using namespace std;

class Numere{
public:
    int *vec;
    int *nr;
    Numere();//definire constructor
    ~Numere();
    void afisare();
};

void Numere::afisare() {
    cout << "n:=" << *nr << endl;
    for(int i=0;i<*nr;i++)
    {
        cout << "vec["<<i<<"]:=" << vec[i] << endl;
    }
}
```

```
Numere::~Numere() {  
    cout << "Apelare destructor!!" << endl;  
    delete this->nr;  
    delete[] this->vec;  
}
```

```
Numere::Numere() {  
    cout << "Initializare obiect!!" << endl;  
    cout << "n:=" << endl;  
    nr=new int;  
    cin >> *nr;  
    vec=new int[200];  
    for(int i=0;i<*nr;i++)  
    {  
        cout << "vec["<<i<<"]:=";  
        cin >> vec[i];  
    }  
}
```

```
class NumereInFisier: public Numere {  
public:  
    void scriefisier()  
    {  
        ofstream myfile ("fisier.txt");  
        if (myfile.is_open())
```

```
{
    for(int i=0;i<*nr;i++)
        myfile << vec[i]<<endl;
    myfile.close();
}
else cout << "Eroare de deschidere fisier!";
}
};
```

```
int main() {
    NumereInFisier obiect;
    obiect.afisare();
    obiect.scriefisier();
    return 0;
}
```

Probleme rezolvate:

a) Modificati programul anterior astfel incat sirul de numere ordonat crescator este scris intr-un fisier.

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

class Numere{
```



```
public:
```

```
    int *vec;
```

```
    int *nr;
```

```
    Numere();//definire constructor
```

```
    ~Numere();
```

```
    void afisare();
```

```
};
```

```
void Numere::afisare() {
```

```
    cout << "n:=" << *nr << endl;
```

```
    for(int i=0;i<*nr;i++)
```

```
    {
```

```
        cout << "vec["<<i<<"]:=" << vec[i] << endl;
```

```
    }
```

```
}
```

```
Numere::~~Numere() {
```

```
    cout << "Apelare destructor!!" << endl;
```

```
    delete this->nr;
```

```
    delete[] this->vec;
```

```
}
```

```
Numere::Numere() {
```

```
cout << "Initializare obiect!!" << endl;

cout << "n:=" << endl;

nr=new int;

cin >> *nr;

vec=new int[200];

for(int i=0;i<*nr;i++)

{

    cout << "vec["<<i<<"]=";

    cin >> vec[i];

}

}
```

```
class NumereInFisier: public Numere {

public:

void scriefisier()

{

    ofstream myfile ("fisier.txt");

    if (myfile.is_open())

    {

        for(int i=0;i<*nr-1;i++)

            for(int j=i+1;j<*nr;j++)

                if (vec[i]>vec[j])

                {

                    int aux=vec[i];

                    vec[i]=vec[j];

                }

            }

    }
```

```
        vec[j]=aux;
    }
    for(int i=0;i<*nr;i++)
        myfile << vec[i]<<endl;
    myfile.close();
}
else cout << "Eroare de deschidere fisier!";
}
};

int main() {
    NumereInFisier obiect;
    obiect.afisare();
    obiect.scriefisier();
    return 0;
}
```

Probleme propuse:

1) Sa se determine probabilitatile de aparitie a numerelor in diferite intervale, considerand un anumit interval de spatiere.

Laborator 11 - Pointeri in C si C++

Problema rezolvata:

Analizati urmatoarele programe in C si C++:

a) Exemplul 1

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8,9,10,11,12,13,14,15,16,17};
    unsigned long int adr0, adr1, adr2,adr3,adr4;
    int *p1, *p2;

    printf("In functie de compilator, in general un byte are 8 biti!\n\n");

    printf("Marimea urmatoarelor variable este:\nsizeof(numar intreg) = %d in bytes\n",
sizeof(int));

    printf("sizeof(numar real) = %d in bytes \n", sizeof(float));

    printf("sizeof(numar unsigned long int) = %d in bytes \n", sizeof(unsigned long int));

    printf("sizeof(numar real double) = %d in bytes \n", sizeof(double));

    printf("\nAdresele din memorie pentru urmatoarele variabile sunt:\na = %p\na + 1 = %p\na + 3
= %p\nna + 300 =%p\n\n", a, a + 1, a + 3,a+300);

    printf("sizeof(a) = %d\n", sizeof(a));

    printf("sizeof(a + 1) = %d\n", sizeof(a + 1));

    printf("sizeof(a + 3) = %d\n\n", sizeof(a + 3));

    printf("sizeof(a + 300) = %d\n\n", sizeof(a + 300));

    adr0 = (unsigned long int) a;
```

```

adr1 = (unsigned long int) (a + 1);
adr2 = (unsigned long int) a + 1;
adr3 = (unsigned long int) (a + 3000);
adr4 = (unsigned long int) a + 3000;

printf("Adresele in numere intregi fara semn sunt:\nadr0 = %lu\nadr1 = %lu\nadr2 =
%lu\n\nadr3 = %lu\nadr4 = %lu\n\n", adr0, adr1, adr2,adr3,adr4);

p1 = a + 1;

printf("\np1 este un pointer la un numar intreg, care va indica la a 2-a locatie de \nmemorie
corespunzatoare sirului de numere intregi");

p2 = &(a[1]);

printf("\nVariabila p1 = %p \nNumarul care este la acea locatie de memorie este: *p1 = %d\n",
p1, *p1);

printf("p2 = %p \t *p2 = %d\n", p2, *p2);

printf("a + 1 = %p \t a[1] = %d\n", a + 1, a[2]);

*p1 = *p1 + 1;

printf("\na[2] = %d\n", a[2]);

return 0;
}

```

b) Exemplul 2

```

#include <iostream>
using namespace std;

int main()
{
    int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8,9,10,11,12,13,14,15,16,17};
}

```

```
unsigned long int adr0, adr1, adr2,adr3,adr4;

int *p1, *p2;

cout<<"Se tine seama de faptul ca un byte are 8 biti!\n\n"<<endl;

cout<<"Marimea urmatoarelor variable este:"<<endl<<"sizeof(numar intreg) =
"<<sizeof(int)<<" in bytes"<<endl;

cout<<"sizeof(numar real) = "<<sizeof(float)<<" in bytes"<<endl;

cout<<"sizeof(numar unsigned long int) = "<<sizeof(unsigned long int)<<" in bytes"<<endl;

cout<<"sizeof(numar real double) = "<<sizeof(double)<<" in bytes"<<endl;

cout<<"Adresele din memorie pentru urmatoarele variabile sunt:"<<endl<<" a =
"<<a<<endl<<" a + 1 = "<<a+1<<endl<<" a + 5 = "<<a+5<<endl;

cout<<"sizeof(a) = "<<sizeof(a)<<endl;

cout<<"sizeof(a + 1) = "<<sizeof(a + 1)<<endl;

cout<<"sizeof(a + 3) = "<<sizeof(a + 3)<<endl;

cout<<"sizeof(a + 300) = "<<sizeof(a + 300)<<endl;

int numar;

int *pointerul;

numar=10;

pointerul=&numar;

int *pointerdoi=pointerul;

cout<<"Numarul de la adresa:"<<pointerul<<" este "<<*pointerul<<endl;

cout<<"Numarul referit de al doilea pointer este:"<<*pointerdoi<<endl;

*pointerdoi=*pointerdoi+2;

cout<<"Numarul referit de al doilea pointer este:"<<*pointerdoi<<endl;

pointerdoi=pointerdoi+2;

cout<<"Numarul referit de al doilea pointer este:"<<*pointerdoi<<endl;

return 0;
```

```
}
```

c) Exemplul 3

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int a,b,aux;
    int *p1, *p2;
    printf("\na:=");
    scanf("%d",&a);
    printf("b:=");
    scanf("%d",&b);
    p1=&a;
    p2=&b;
    aux=*p1;
    *p1=*p2;
    *p2=aux;
    printf("\na:=%d",a);
    printf("\nb:=%d",b);
    return 0;
}
```

d) Se citeste un vector de N numere de la tastatura. Sa se interschimbe numarul maxim cu numarul minim.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int aux;
    int *p1, *p2;
    int i,n,v[200],minim=999,maxim=-999, indicemin,indicemax;
    printf("\nn:=");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nv[%d]:=",i);
        scanf("%d",&v[i]);
    }
    for(i=0;i<n;i++)
    {
        if (minim>v[i])
        {
            indicemin=i;
            minim=v[i];
        }
    }
}
```



```
    if (maxim<v[i])
    {
        indicemax=i;
        maxim=v[i];
    }
}

printf("Minim gasit la locatia %d cu valoarea %d\n",indicemin,v[indicemin]);
printf("Maxim gasit la locatia %d cu valoarea %d\n",indicemax,v[indicemax]);
p1=v+indicemax;
p2=v+indicemin;
aux=*p1;
*p1=*p2;
*p2=aux;
for(i=0;i<n;i++)
{
    printf("\nv[%d]:=%d\n",i,v[i]);
}
return 0;
}
```

e) Sa se scrie sortarea unui vector citit de la tastatura, folosind pointeri.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <math.h>

int main()
{
    int aux;

    int *p1, *p2;

    int i,j,n,v[200];

    printf("\nn:=");

    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("\nv[%d]:=",i);

        scanf("%d",&v[i]);

    }

    for(i=0;i<n-1;i++)

        for(j=i+1;j<n;j++)

            if(v[i]<v[j])

                {

                    p1=v+i;

                    p2=v+j;

                    aux=*p1;

                    *p1=*p2;

                    *p2=aux;

                }

    for(i=0;i<n;i++)

    {
```

```
    printf("\nv[%d]:=%d\n",i,v[i]);  
}  
return 0;  
}
```

Probleme propuse:

- 1) Sa se construiasca o clasa care sa contina subprogramele de mai sus in functiile sale membre sau in constructor.
- 2) Sa se scrie realizeze interclasarea a doua siruri de numere reale citite de la tastatura, folosind pointeri.

Limbaje de programare |

Laborator 12 - Pointeri cu functii

Probleme rezolvate

a) Analizati urmatorul program din cadrul laboratorului precedent:

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()
{
    int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8,9,10,11,12,13,14,15,16,17};
    unsigned long int adr0, adr1, adr2,adr3,adr4;
    int *p1, *p2;

    printf("In functie de compilator, in general un byte are 8 biti!\n\n");

    printf("Marimea urmatoarelor variable este:\nsizeof(numar intreg) = %d in bytes\n",
sizeof(int));

    printf("sizeof(numar real) = %d in bytes \n", sizeof(float));

    printf("sizeof(numar unsigned long int) = %d in bytes \n", sizeof(unsigned long int));

    printf("sizeof(numar real double) = %d in bytes \n", sizeof(double));

    printf("\nAdresele din memorie pentru urmatoarele variabile sunt:\na = %p\na + 1 = %p\na + 3
= %p\nna + 300 =%p\n\n", a, a + 1, a + 3,a+300);

    printf("sizeof(a) = %d\n", sizeof(a));

    printf("sizeof(a + 1) = %d\n", sizeof(a + 1));

    printf("sizeof(a + 3) = %d\n\n", sizeof(a + 3));

    printf("sizeof(a + 300) = %d\n\n", sizeof(a + 300));

    adr0 = (unsigned long int) a;

    adr1 = (unsigned long int) (a + 1);
```

```

adr2 = (unsigned long int) a + 1;
adr3 = (unsigned long int) (a + 3000);
adr4 = (unsigned long int) a + 3000;

printf("Adresele in numere intregi fara semn sunt:\nadr0 = %lu\nadr1 = %lu\nadr2 =
%lu\n\nadr3 = %lu\nadr4 = %lu\n\n", adr0, adr1, adr2,adr3,adr4);

p1 = a + 1;

printf("\np1 este un pointer la un numar intreg, care va indica la a 2-a locatie de \nmemorie
corespunzatoare sirului de numere intregi");

p2 = &(a[1]);

printf("\nVariabila p1 = %p \nNumarul care este la acea locatie de memorie este: *p1 = %d\n",
p1, *p1);

printf("p2 = %p \t *p2 = %d\n", p2, *p2);

printf("a + 1 = %p \t a[1] = %d\n", a + 1, a[2]);

*p1 = *p1 + 1;

printf("\na[2] = %d\n", a[2]);

return 0;
}

```

b) Sa se scrie o functie care returneaza un pointer catre numarul maxim dintr-un vector de numere intregi.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int* functie(int v[],int n)
{
    int i,maxim=-9999,indice;

```

```
int *p;
for(i=0;i<n;i++)
    if (v[i]>maxim)
    {
        maxim=v[i];
        indice=i;
    }
p=v+indice;//sau p=&v[indice]
return p;
}

int main()
{
    int i,n,v[200];
    int *p;
    printf("\nn:=");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nv[%d]:=",i);
        scanf("%d",&v[i]);
    }
    p=functie(v,n);
    printf("Pointerul este:=%p, iar numarul este:=%d",p,*p);
    return 0;
}
```

```
}
```

c) Sa se scrie o functie care returneaza un pointer catre numarul minim dintr-un vector de numere intregi.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

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

```
int* functie(int v[],int n)
```

```
{
```

```
    int i,minim=9999,indice;
```

```
    int *p;
```

```
    for(i=0;i<n;i++)
```

```
        if (v[i]<minim)
```

```
        {
```

```
            minim=v[i];
```

```
            indice=i;
```

```
        }
```

```
        p=v+indice;//sau p=&v[indice]
```

```
        return p;
```

```
}
```

```
int main()
```

```
{
```

```
int i,n,v[200];

int *p;

printf("\nn:=");

scanf("%d",&n);

for(i=0;i<n;i++)
{
    printf("\nv[%d]=",i);
    scanf("%d",&v[i]);
}

p=functie(v,n);

printf("Pointerul este:=%p, iar numarul este:=%d",p,*p);

return 0;
}
```

d) Sa se scrie o functie care realizeaza interschimbarea a doi vectori, folosind pointeri.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void functie(int* v,int* u,int n)
{
    int i,aux;

    int *pv, *pu;
```



```
pv=v;
pu=u;
for(i=0;i<n;i++)
{
    aux=*pv;
    *pv=*pu;
    *pu=aux;
    pv=pv+1;
    pu=pu+1;
}
}

int main()
{
    int i,n,v[200],u[200];
    printf("\nn:=");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nv[%d]:=",i);
        scanf("%d",&v[i]);
    }
    for(i=0;i<n;i++)
    {
        printf("\nu[%d]:=",i);
```

```
    scanf("%d",&u[i]);
}
functie(v,u,n);
for(i=0;i<n;i++)
{
    printf("\nv[%d]:=%d",i,v[i]);

}
for(i=0;i<n;i++)
{
    printf("\nu[%d]:=%d",i,u[i]);
}
return 0;
}
```

e) Sa se scrie o functie care realizeaza interschimbarea a doua matrici patratice, folosind pointeri.

Obs! Urmatorul program functioneaza doar daca se alocă memorie statica corecta; de exemplu pentru o dimensiune $n=3$ sa se aloce $v[3][3]$ si $u[3][3]$, altfel programul genereaza eroare datorita parcurgerii suplimentare a memoriei ne-alocate.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

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

```
void functie(int* v,int* u,int n)
```

```
{
    int i,j,aux;
    int *pv, *pu;
    pv=v;
    pu=u;
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            {
                aux=*pv;
                *pv=*pu;
                *pu=aux;
                pv=pv+1;
                pu=pu+1;
            }
}

int main()
{
    int j,i,n,v[3][3],u[3][3];
    printf("\nn:=");
    scanf("%d",&n);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            {
                printf("\nv[%d][%d]:=",i,j);
```

```

        scanf("%d",&v[i][j]);
    }
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("\nu[%d][%d]:=",i,j);
            scanf("%d",&u[i][j]);
        }
    functie(v,u,n);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("\nv[%d][%d]:=%d",i,j,v[i][j]);
        }

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("\nu[%d][%d]:=%d",i,j,u[i][j]);
        }
    return 0;
}

```

f) Versiunea corecta cu alocare dinamica de memorie!

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

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

```
int functie(int *v,int *u,int n)
```

```
{
```

```
    int i,j,aux;
```

```
    for(i=0;i<n;i++)
```

```
        for(j=0;j<n;j++)
```

```
        {
```

```
            printf("%d",i);
```

```
            aux=*v;
```

```
            *v=*u;
```

```
            *u=aux;
```

```
            v=v-1;
```

```
            u=u-1;
```

```
        }
```

```
    return 1;
```

```
}
```

```
int main()
```

```
{
```

```
    int j,i,nr,*v,*u,n;
```

```
/*daca se aloca static memorie programul nu functioneaza, deoarece for-urile se fac cu  
elementele incluse*/
```

```
printf("\nn:=");
```

```
scanf("%d",&n);
```

```
v=(int*) malloc ((n*n)*sizeof(int));
```

```
u=(int*) malloc ((n*n)*sizeof(int));
```

```
for(i=0;i<n;i++)
```

```
    for(j=0;j<n;j++)
```

```
{
```

```
    printf("\nv[%d][%d]:=",i,j);
```

```
    scanf("%d",&nr);
```

```
    *v=nr;
```

```
    v=v+1;
```

```
}
```

```
v=v-1;
```

```
for(i=0;i<n;i++)
```

```
    for(j=0;j<n;j++)
```

```
{
```

```
    printf("\nu[%d][%d]:=",i,j);
```

```
    scanf("%d",&nr);
```

```
    *u=nr;
```

```
    u=u+1;
```

```
}
```

```
u=u-1;
```

```
functie(v,u,n);

for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        printf("\nv[%d][%d]:=%d",i,j,*v);
        v=v-1;
    }

for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        printf("\nu[%d][%d]:=%d",i,j,*u);
        u=u-1;
    }
return 0;
}
```

Probleme propuse:

1) Sa se transcrie programele precedente in C++ si sa se construiasca o structura de clasa pe baza acestora cu functii membre, constructori si destructori.

Laborator 13 - Siruri de caractere, variabile char si string

In C, un sir de caractere este un vector de elemente de tip char, in care ultimul caracter este '\0' (cod ASCII 0).

Analizati urmatorul program:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("Introduceti un sir de la tastatura fara spatii! \n");
    char word[64];
    scanf("%s", word);
    printf("%s\n",word);
    printf("\nIntroduceti un sir de la tastatura cu spatii! \n");
    fflush(stdin);//curatare streaming in input
    scanf("%[^\n]s",word);
    /*
    [] este un caracter de tipul scanset. [^\n] citeste cat timp nu este prezent caracterul de linie
    noua ('\n') .
    */
    printf("%s",word);
    return 0;
}
```


Probleme rezolvate:

a) Folosind functii si pointeri, sa se determine numarul de caractere dintr-un sir de caractere(inclusiv spatii) citit de la tastatura.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int functie(char *p)
```

```
{
```

```
    int i=0;
```

```
    while(*p!='\0')
```

```
    {
```

```
        p=p+1;
```

```
        i=i+1;
```

```
    }
```

```
    return i;
```

```
}
```

```
int main()
```

```
{
```

```
    printf("Introduceti un sir de la tastatura fara spatii! \n");
```

```
    char word[64];
```

```
    scanf("%s", word);
```

```
    printf("%s\n",word);
```

```
    printf("%s",word);
```

```
printf("Numarul de caractere este:=%d",functie(word));  
printf("\nIntroduceti un sir de la tastatura cu spatii! \n");  
fflush(stdin);  
scanf("%[^\\n]s",word);  
  
/*  
[] este un caracter de tipul scanset. [^\\n] citeste cat timp nu este prezent caracterul de linie  
noua (\\n) .  
*/  
  
printf("%s",word);  
printf("Numarul de caractere este:=%d",functie(word));  
return 0;  
}
```

Probleme propuse:

- 1) Folosind functii si pointeri, sa se determine daca un sir de caractere(inclusiv spatii) este palindrom.
- 2) Folosind functii si pointeri, sa se determine numarul de spatii dintr-un sir de caractere citit de la tastatura.
- 3) Folosind pointeri si functii, sa se converteasca un sir de caractere intr-un numar intreg.

Laborator 14 – Metoda Euler de rezolvare a ecuatiilor diferentiale ordinare

Exemplu:

```
#include<stdio.h>

int main()
{
    float a,b,x,y,h,t,k;

    float functie;

    printf("\nConditile initiale [x0,y0]: ");
    scanf("%f%f",&a,&b);

    printf("\nIntervalul de spatiere h: ");
    scanf("%f",&h);

    printf("\nTimpul maxim de evaluare [0,t]: ");
    scanf("%f",&t);

    x=a;
    y=b;

    printf("\n x\t y\n");
    while(x<=t)
    {
        functie=-y+x+20;
        k=h*functie;
        y=y+k;
        x=x+h;

        printf("%0.3ft%0.3fn",x,y);
    }
}
```

```
return 0;  
}
```

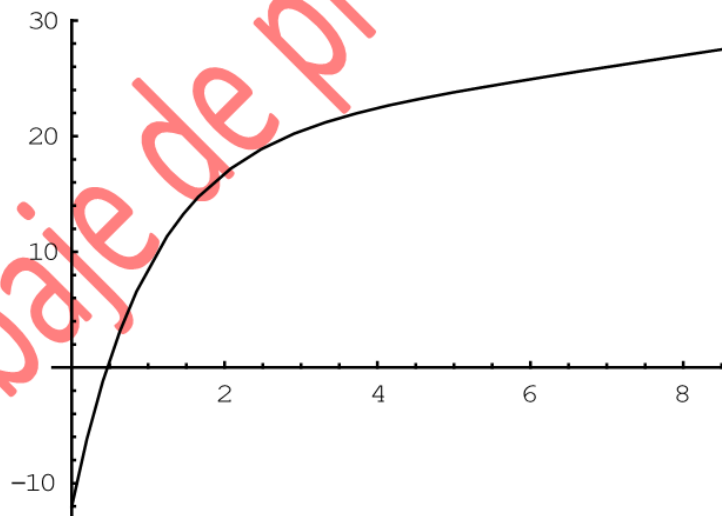
Probleme propuse:

- 1) Sa se determine ecuatia diferentiala care este analizata in programul precedent.
- 2) Sa se modifice programul anterior pentru a rezolva toate sistemele de ecuatii diferentiale propuse mai jos.
- 3) Rescrieti programul anterior folosind vectori si functii.
- 4) Sa se determine conditia initiala y_0 corespunzatoare x_0 astfel incat la $t=10$ solutia numerica sa fie egala cu y_f , pentru diverse valori ale pasului de integrare h .

Ecuatii diferentiale in Wolfram Mathematica

```
H=DSolve[{u'[x]==-u[x]+x+20,u[0]==-12},u[x],x];
```

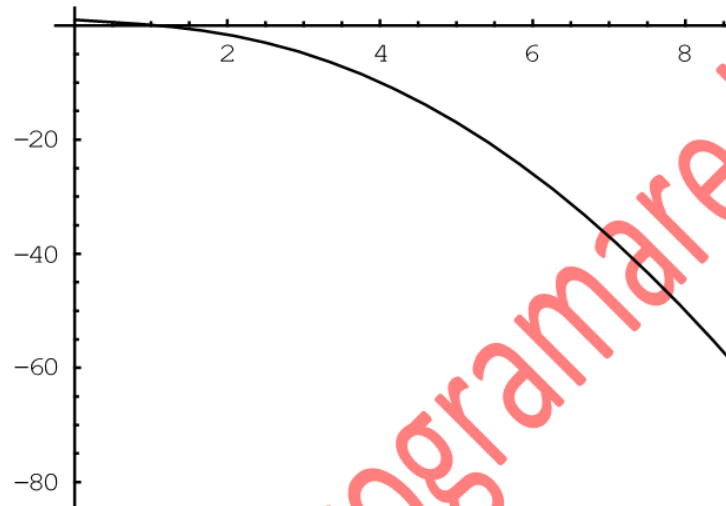
```
Plot[u[x]/.H[[1]],{x,0,10}]
```



- Graphics -

```
H=DSolve[{u'[x]==- x*x-u[x],u[0]==1},u[x],x];
```

```
Plot[u[x]/.H[[1]],{x,0,10}]
```

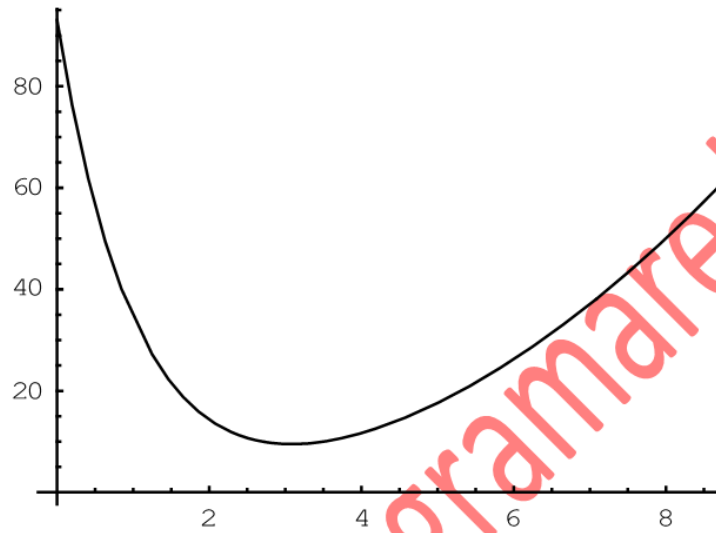


- Graphics -

Limbaje de programare I

```
H=DSolve[{u'[x]== x*x-u[x],u[0]==93},u[x],x];
```

```
Plot[u[x]/.H[[1]],{x,0,10}]
```



- Graphics -

Limbaje de programare I