PROIECT LA INFORMATICA

**Subprograme**

Elev:

* Frinca Fernando

**Implementarea subprogramelor in limbajul C++**

**1.Notiuni introductive**

Blocul este unitatea de baza a oricarui program C++.

Acesta este incapsulat intr-o instructiune compusa si este delimitat de {…}.

Orice program contine un bloc principal, numit functie radacina( int main() ).

Un bloc este compus din doua categorii de instructiuni( instructiuni declarative si intructiuni procedurale sau executabile ). Blocul are forma:

Instructiuni

procedurale

Instructiuni declarative

antet

{corp}

Studiu de caz:

int main() → antet

{

int a,b;

float ma;

cout<<”a=”;

cin>>a;

cout<<”b=”;

cin>>b;

ma=(float)(a+b)/2;

cout<<”Media=”<<ma;

return 0;

}

Instructiuni declarative

Corp

Instructiuni procedurale

Definitie!

Subprogramul este o secventa de instructiuni ce rezolva o anumita sarcina si care poate fi scrisa separat de blocul radacina si lansata in executie din interiorul oricarui bloc, ori de cate ori este nevoie.

Utilitatea subprogramelor intervine atunci cand o problema poate fi descompusa in mai multe subprobleme( numite module ) sau cand o secventa de instructiuni se repeta in acelasi program.

Intr-un program exista doua categorii de module:

* modulul apelant: este cel care pentru rezolvarea propriilor sarcini apeleaza la alt modul( modulul apelat );
* modulul apelat: este cel care rezolva sarcinile primite de la modulul apelant.

Modulul apelant(P)

* instr. 1
* instr. 2

.

.

.

* instr. i

.

.

.

* instr. j

.

.

.

* instr. n

Modulul apelat(SP)

Revenirea

din control

Transfera  
  
controlul

apel

Comunicarea intre module se realizeaza prin intermediul parametrilor.

Parametrii sunt de trei tipuri:

* parametrii de intrare( sunt transmisi de catre programul principal subprogramului, pot fi prelucrati doar in intermediul subprogramului SP). P → SP
* parametrii de iesire( sunt transmisi de catre subprogram, programului principal; el reprezinta de fapt rezultatele, pot fi prelucrati doar de catre programul principal P). SP → P
* parametrii de intrare-iesire, au dublu rol si pot fi prelucrati atat in interiorul programului P cat si in interiorul subprogramului SP. P ↔ SP

Observatie!

In limbajul C++ subprogramele se mai numesc functii.

Elementele unui subprogram sunt:

* definitia subprogramului, specifica ceea ce realizeaza subprogramul
* apelul subprogramului, reprezinta lansarea in executie a subprogramului
* prototipul subprogramului, ofera compilatorului informatii despre subprogram

Studiu de caz

#include<iostream>

using namespace std;

void scrie(); prototipul

int main()

{

scrie(); apel

return 0;

}

void scrie() antet

{

cout<<”Hello C++!”; corp definitie

}

Observatie!

In cazul in care definitia subprogramului este scrisa inaintea de blocul in care acesta este apelata, prototipul poate sa lipseasca. Deoarece antetul din definitia subprogramului are rol declarativ.

#include<iostream>

using namespace std;

void scrie()

{

cout<<”Hello C++!”;

}

int main()

{

scrie();

return 0;

}

**2.Clasificarea subprogramelor**

Exista 2 criterii de clasificare:

1. Dupa autor:
2. Subprograme standard( de sistem )

* sunt cele predefinite in limbajul de programare si au fost create chiar de autorii limbajului

Ex: sqrt(x), pow(x,n), strlen(s), strcpy(d,s)

* functiile utilizate in prelucrarea sirurilor de caractere, functii de citire si afisare, etc;

Observatie!

Aceste functii sunt definite in fisierele header

care trebuie incluse in program utilizand directive de

preprocesare.

#include<fisier\_header>

1. Subprograme utilizator( nestandard )

* sunt cele create de catre programatori

Ex: scrie()

1. Dupa modalitatea de apelare:

1)Subprograme procedurale

- care sunt apelate utilizand sintaxa:

nume\_subprogram(lista\_parametrii)

Observatii!

1.Functiile procedurale au tipul void ceea ce inseamna ca nu returneaza nimic prin numele lor. Ele pot returna rezultate prin intermediul parametrilor.

2.Lista de parametrii poate fi vida.

3.Este posibil ca functiile procedurale sa nu returneze rezultate nici prin intermediul parametrilor.

2) Subprogramele operand,

sunt apelate astfel:

- utilizand o instructiune de afisare, de exemplu:

cout<<nume\_subprogram(lista\_parametrii);

-intr-o operatie de atribuire, de exemplu:

int n=strlen(s);

-intr-o expresie, de exemplu:

s=a+nume\_subprogram(lista\_parametrii);

s=a+pow(x,n);

Observatie!

Subprogramele operand returneaza un rezultat prin chiar numele lor si pot returna si alte rezultate prin intermediul parametrilor. Ele au un tip de date elementare( int, float, unsigned ), iar rezultatul returnat de ele va fi de acest tip.

Sintaxele definitiilor subprogramelor procedurale si operand:

* Subprogramele procedurale:

void nume\_subprogram(lista\_parametrii)

{

instructiuni declarative

corp

instructiuni procedurale

}

* Subprograme operand:

tip\_data nume\_subprogram(lista\_parametrii)

{

instructiuni declarative

corp instructiuni procedural

return expresie;

}

**Reguli pentru construirea subprogramelor C++**

Definitia subprogramului

Definitia unui subprogram este format din antetul si corpul subprogramului:

<antetul subprogramului>

{

<declaratii proprii subprogramului>

<instructiuni>

[return<expresie>;]

}

1. Antetul subprogramului este o linie de recunoastere a subprogramului, in care i se atribuie un nume. El specifica inceputul subprogramului.
2. Corpul subprogramului, la fel ca orice bloc C++, este incapsulat intr-o instructiune compusa, delimitate de acolade si este format din doua parti:

* partea declarativa contine definitii de elemante folosite numai in interiorul subprogramului: tipuri de date, constant si variabile de memorie. Nu se pot definii si alte subprograme(nu este variabila tehnica de imbricare a subprogramelor existent in alte limbaje de programare).
* Partea executabila contine instructiunile prin care sunt descrise actiunile realizate de subprogram.

**Antetul subprogramului**

Subprogramul trebuie sa aiba un antet prin care se precizeaza interfata dintre programul apelat si subprogram. El contine 3 categorii de informatii:

* Tipul rezultatului pentru functiile operand se precizeaza tipul furnizat de subprogram prin chiar numele sau. Pentru functiile procedural, tipul rezultatului este void( nu intoarce nici un rezultat prin numele sau; rezultatele vor fi intoarse prin parametrii subprogramului). Daca nu se precizeaza tipul rezultatului, compilatorul va considera ca acesta este implicit de tip int.
* Numele subprogramului este un identificator unic, care se atribuie subprogramului. Numele trebuie sa respecte aceleasi reguli ca orice identificator C++.
* Parametrii folositi pentru comunicare; pentru fiecare parametru se precizeaza numele si tipul.

Antetul unui subprogram este de forma:

tip rezultat nume\_subprogram(lista\_parametrii)

Lista de parametrii este de forma:

(tip 1 p1, tip 2 p2, tip 3 p3,…, tip n pn)

separator

tipul parametrului identificatorul parametrului

Exemplu 1: float alfa(int a, int b, float c)

Acesta este antetul unei functii operand care furnizeaza un rezultat de tip float. Numele functiei este alfa, iar parametrii folositi pentru comunicare sunt a si b de tip int si c de tip float.

Exemplul 2: void beta(int a, float b, float c, char d)

Exemplul 3: void gama()

Observatii!

1. Separarea parametrilor in lista se face prin “,”.Daca exista mai multi parametrii de acelasi tip, ei nu pot fi grupati ca la declararea tipului variabilelor de memorie. Pentru fiecare parametru trebuie precizat tipul.
2. Tipul parametrilor poate fi:

* Orice tip standard al sistemului folosit pentru datele elementare- intreg(int, unsigned, long), real(double, float, long double) sau caracter(char sau unsigned char), -tipul pointer sau orice tip de structura de data( vector, matrice, sir de caractere sau inregistrare)
* Orice tip definit de utilizator inainte de a define subprogramul.

1. Pentru rezultatul functiei nu se poate folosi tipul tablou de memorie.

Exemplu: float a[10] tablou(int v, unsigned n)

Acest antet de subprogram va produce eroare deoarece tipul functiei este tablou de memorie.

1. Numele subprogramului poate fi folosit in trei locuri distincte:

* In prototipul subprogramului, unde are un rol declarative;
* In antetul subprogramului, unde are un rol de definitie, dar si declarative;
* In apelarea subprogramului, unde are rol de activare.

**Corpul subprogramului**

Corpul subprogramului este un bloc care contine atat instructiuni declarative, cat si instructiuni imperative. Variabilele de memorie declarate in corpul subprogramului se numesc variabile locale. In cazul unei functii operand, ultima instructiune din corpul subprogramului trebuie sa fie instructiunea return, care are sintaxa:

return<expresie>;

Valoarea obtinuta prin evaluarea expresiei <expresie> va fi atribuita functiei operand( va fi valoarea returnata prin numele functiei). Rezultatul expresiei trebuie sa fie de acelasi tip cu tipul functiei sau de un tip care poate fi convertit implicit in tipul functiei.

Cand compilatorul C++ intalneste intr-un subprogram instructiunea return, termina executia subprogramului si reda controloul modulului apelant. Prin urmare, daca veti scrie in subprogram dupa instructiunea return alte instructiuni, ele nu vor fi executate.

**Prototipul subprogramului**

Este o linie de program, aflata inaintea modulului care apeleaza subprogramul, prin care se comunica compilatorului informatii despre subprogram( se declara subprogramul). Prin declararea programului, compilatorul primeste informatii despre modulul in care se poate apela subprogramul si poate face verificari la apelurile de subprogram in ceea ce priveste tipul parametrilor folositi pentru comunicare si a modulului in care poate face conversia acestor parametrii.

Un subprogram, pentru a putea fi folosit, trebuie declarat. Pentru declararea lui se foloseste prototipul. El contine 3 categorii de informatii, la fel ca si antetul subprogramului: tipul rezultatului, numele subprogramului si tipul parametrilor folositi pentru comunicare. Pentru fiecare parametru din antetul subprogramului, se pot preciza numai tipul, nu si numele lui.

Prototipul unui subprogram este de forma:

tip rezultat nume\_subprogram(lista \_tipuri\_parametrii)

Lista tipurilor parametrilor este de forma:

tip 1, tip 2, tip 3, …, tip n

Observatii:

1. Separarea tipurilor de parametrii in lista se face prin caracterul “,”. Lista trebuie sa contina atatea tipuri de parametrii cati parametrii au fost definite in antetul subprogramului. In lista se precizeaza tipul de data la care se refera, in aceeasi ordine in care au fost scrisi parametrii la definirea lor in antet.
2. Spre deosebire de antetul subprogramului , prototipul se termina cu caracterul “;”.

Pentru functiile al caror antet a fost precizat anterior, prototipurile vor fi:

Exemplul 1: float alfa(int , int, float);

Exemplul 2: void beta(int , float, float, char);

Exemplul 3: void game();

**Activarea (apelul) subprogramului**

Subprogramul trebuie sa fie cunoscut atunci cand se cere prin apel activarea lui.

* Daca subprogramul este standard, trebuie inclus fisierul care contine prototipul subprogramului in fisierul sursa.
* Daca subprogramul este utilizator, trebuie declarat fie prin folosirea prototipului,fie prin definirea lui inaintea apelului.

In functie de modul in care a fost definit, subprogramul se activeaza fie printr-o instructiune procedural, fie ca operand intr-o expresie.

Pentru functiile ale caror antet a fost precizat anterior, activarea se poate face astfel:

Exemplul 1: w=alfa(x,y,z);

Exemplul 2: int x; float y,z; char w;

beta(x,y,z,w);

Exemplul 3: gama();

Atentie!

Orice subprogram trebuie declarat si definit. Declararea unui

subprogram este necesara pentru ca el sa fie cunoscut de subprogram care il apeleaza. Declararea lui poate fi facuta fie prin prototip, fie prin definitia lui (antetul impreuna cu corpul subprogramului). Pentru a declara subprogramul, fie scrieti prototipul inaintea subprogramului care il apeleaza, putand scrie apoi definitia lui oriunde in program, fie definiti subprogramul inaintea subprogramului care il apeleaza.

#include<iostream>

void scrie() antetul subprogramului

{ definitia subprogramului

cout<<”Subprogram”; corpul subprogramului

}

Modul apelant main()

void main()

{

Modul apelat scrie()

scrie(); apelul subprogramului

}

Asadar:

* Prototipul subprogramului declara subprogramul
* Apelul subprogramului executa subprogramul
* Antetul subprogramului specifica numele subprogramului si tipul argumentelor si al valorilor returnate, iar corpul subprogramului il defineste, adica specifica ceea ce trebuie sa realizeze subprogramul.

**Parametrii de comunicare**

Daca pentru comunicarea intre subprogram si bloc apelant se folosesc parametrii,acestia vor fi scrisi dupa numele subprogramului, intre paranteze rotunde, astfel:

* In antet, pentru fiecare parametru se precizeaza denumirea simbolica folosita in interiorul subprogramului. Acesti parametrii se numesc parametrii formali. In prototip, pentru fiecare parametru se precizeaza tipul de data la care se refera, in aceeasi ordine in care au fost scrisi la definirea lor in antet.
* La activarea subprogramului, parametrilor de comunicare li se vor atribui valori concrete cu care se va executa subprogramul la acest apel. Aceste valori vor fi communicate la apelul subprogramului, dupa numele subprogramului, intre paranteze rotunde, in aceeasi ordine in care au fost scrisi la definirea lor in antet. Comunicarea trebuie sa respecte aceeasi succesiune, tipuri de date si numar de parametrii ca si in lista de parametrii formali, deoarece atribuirea valorilor se face respectand regula de corespondenta. Acesti parametrii se numesc parametrii actuali.

Observatii!

1. Numele parametrilor actuali poate fi diferit de numele parametrilor formali.
2. Tipul parametrilor formali poate sa fie diferit de tipul parametrilor actuali numai cand parametrii actuali pot fi convertiti la tipul celor formali.

**Transferul parametrilor intre subprograme**

Transferul parametrilor reprezinta tehnica utilizata pentru comunicarea datelor intre module.

Exista urmatoarele metode de transfer:

1. Transferul prin valoare:

* In momentul apelarii subprogramului valoarea parametrilor este incarcata in stiva sistemului
* Acesti parametrii se numesc parametrii valoare.
* Valoarea lor din stiva este accesibila doar modulului apelat pe parcursul executiei acestuia.
* Dupa incheierea executiei modulului apelat valoarea din stiva se pierde.
* Aceasta metoda de transfer poate fi utilizata doar pentru parametriii de intrare.

Exemplu: void suma(int a, int b)

{…}

* Pentru a utilize transferul prin valoare pentru parametrii de iesire sau pentru cei de intrare-iesire se utilizeaza variabilele de tip pointer.

Exemplu: void suma(int a, int b, int \*s)

{…}

suma (a,b,&s);

* Astfel pentru transmiterea rezultatelor din subprogram catre programul principal se utilizeaza transferul prin valoare utilizand pointer.

1. Transferul prin referinta( adresa )

* Parametrii transferati prin aceasta metoda se numesc parametrii variabila.
* Parametrii pastreaza adresa din stiva astfel ca valoarea acestora este reflectata si dupa incheierea executiei subprogramului si in programul principal; aceasta tehnica se foloseste pentru transmiterea parametrilor(iesire, intrare-iesire)

Exemplu: void suma( int a, int b, int &s)

{…}

suma(a,b,s);

**Utilizarea stivei de catre subprograme**

In memoria interna exista o zona anumita stiva sistemului( stack ) in care se pastreaza temporar informatii despre subprogram. Aceste informatii formeaza instant subprogramului care este formata din:

Instanta subprogramului

Parametrii locali

Contractual de apel

Adresa de revenire

Parametrii de comunicare

Adresa de revenire reprezinta adresa instructiunii din modulul apelant care urmeaza imediat dupa apelul subprogramului.

Etapele execuatate la apelul subprogramului sunt:

1. Se preda controlul modulului apelat
2. Se pregateste stiva sistemului astfel:

* Se salveaza/incarca in stiva adresa de revenire
* Se incarca valorile parametrilor( de la dreapta la stanga )
* Se rezerva spatiu pentru variabilele locale.

1. Se executa instructiunile din corpul modulului apelat.

Exemplu: alfa(x,y,z)

Stiva:

|  |
| --- |
| Variabile locale |
| x |
| y |
| z |
| Adresa de revenire |

Etapele executate la incheierea executarii programului sunt:

1. Se intrerupe executia modulului apelat
2. Se elibereaza stiva sistemului:

* Se elibereaza spatial ocupat de parametrii
* Se extrage adresa de revenire

1. Se reda controlul modulului apelant si se continua executia acestuia cu instructiunea de la adresa de revenire.

**Tablourile de memorie si subprogramele**

Citirea si afisarea vectorului cu ajutorul subprogramelor:

void citire( unsigned n, int v[100])

{ unsigned i;

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

cin>>v[i];

}

void afisare( unsigned n, int v[100])

{ unsigned i;

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

cout<<v[i]<<” “;

}

int main()

{ int x[100],y[100]; unsigned n;

cin>>n; citire(n,x); citire(n,y); afisare(n,x); afisare(n,y);

return 0;

}

Citirea si afisarea unei matrice cu ajutorul subprogramelor:

void citire( unsigned n, unsigned m, int a[20][30])

{ unsigned i,j;

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

for(j=1;j<=m;j++)

cin>>a[i][j];

}

void afisare( unsigned n, unsigned m, int a[20][30])

{ unsigned i,j;

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

{ for(j=1;j<=m;j++)

cout<<a[i][j]<<” “;

cout<<endl;}

}

int main()

{ int x[20][30],y[20][30]; unsigned n,m;

cin>>n; cin>>m; citire(n,m,x); citire(n,m,y); afisare(n,m,x); afisare(n,m,y);

return 0;

}

**Clasificarea variabilelor de memorie**

Exista in memoria interna urmatoarele zone in care se poate rezerva spatiu pentru variabile:

* Stiva sistemului (stack )
* Segmental de date (data segment )
* Zona de adrese libere ( heap )

Variabilele de memorie pot fi clasificate dupa urmatoarele doua criterii:

1. Durata de viata reprezinta perioada de timp in care variabilelor li se aloca spatiu in memoria interna.
2. Domeniul de vizibilitate reprezinta zona din program in care este permis accesul la variabile.
3. Dupa durata de viata:
4. variabile cu durata locala:

* Sunt cele declarate in interiorul unui bloc ( subprogram )
* Li se rezerva spatiu in stiva, primind la declarare valoarea reziduala( valoarea care se gaseste in stiva in locul in care a fost declarata variabila ).
* Durata lor de viata coincide cu perioada in care se executa instructiunile din corpul subprogramului.
* Nu pot fi accesate inafara blocului in care au fost declarate.

1. Variabile cu durata static:

* Sunt declarate inafara oricarui bloc
* Pot fi accesate in toate subprogramele definite fupa declararea lor.
* Li se rezerva spatiu in segmental de date si sunt initializate in momentul declararii cu valoarea 0.

1. Variabile cu durata dinamica:

* Li se rezerva spatiu doar in momentul in care sunt utilizati.
* Zona de memorie folosita de aceste variabile este zona de adrese libere.
* Imediat dupa utilizarea acestora zona de memorie a acestora este eliberata.
* Sunt variabile care contin adrese si se numesc pointeri.
* Utilizarea variabilelor dinamice reprezinta cea mai eficienta metode de alocare a memoriei.

1. Dupa domeniul de vizibilitate:
2. Variabile locale( corespund cu variabilele cu durata locala de la primul criteriu ).
3. Variabile globale (corespund variabilelor cu durata static ).

Observatie:

In cazul in care intr-un program exista o variabila locala si o variabila globala cu aceeasi nume, in subprogramul in care a fost declarata cea locala are intaietate.

Cuprins:

**1.Partea teoretica.**

* 1.1.Notiuni introductive
* 1.2.Clasificarea subprogramelor
* 1.3.Reguli pentru construirea subprogramelor C++
* 1.4.Utilizarea stivei de catre subprograme
* 1.5.Tablourile de memorie si subprogramele
* 1.6.Clasificarea variabilelor de memorie

**2.Prezentarea** [**aplicatiei**](Proiect%20subprograme.cpp)**.**

* 2.1.Pentru programele implementate cu ajutorul subprogramelor primite la tama.
* 2.2.Pentru alte programe implementate cu ajutorul subprogramelor.

#include<iostream>

#include<math.h>

#include<fstream>

using namespace std;

void citire(int &n)

{

cin>>n;

}

void afisare(float n)

{

cout<<n<<endl;

}

void suma(int x, int y, int &sn)

{

sn=x+y;

}

void diferenta(int x, int y, int &dn)

{

dn=x-y;

}

void produs(int x, int y, int &pn)

{

pn=x\*y;

}

void cat(int x, int y, float &cn)

{

cn=(float)x/y;

}

void citire1(unsigned &n)

{

cin>>n;

}

void afisare1(float n)

{

cout<<n<<endl;

}

void suma1(unsigned x, unsigned y, unsigned &sn)

{

sn=x+y;

}

void diferenta1(unsigned x, unsigned y, int &dn)

{

dn=x-y;

}

void produs1(unsigned x, unsigned y, unsigned &pn)

{

pn=x\*y;

}

void cat1(unsigned x, unsigned y, float &cn)

{

cn=(float)x/y;

}

struct complex

{

float re;

float im;

};

complex z1,z2,zs,zd,zp,zr;

void citire2(complex &z)

{

cin>>z.re;

cin>>z.im;

}

void afisare2(complex &z)

{

cout<<z.re<<"+"<<z.im<<"i"<<endl;

}

void produs2(complex z1, complex z2, complex &zs)

{

zp.re=z1.re\*z2.re-z1.im\*z2.im;

zp.im=z1.re\*z2.im+z1.im\*z2.re;

}

void raport2(complex z1, complex z2, complex &zr)

{

zr.re=(z1.re\*z2.re+z1.im\*z2.im)/(z2.re\*z2.re+z2.im\*z2.im);

zr.im=(-z1.re\*z2.im+z1.im\*z2.re)/(z2.re\*z2.re+z2.im\*z2.im);

}

void suma2(complex z1, complex z2, complex &zs)

{

zs.re=z1.re+z2.re;

zs.im=z1.im+z2.im;

}

void diferenta2(complex z1, complex z2, complex &zd)

{

zd.re=z1.re-z2.re;

zd.im=z1.im-z2.im;

}

struct fractie

{

int numarator;

int numitor;

};

fractie f,f1,f2,fp,fr,fs,fd;

void citire3(fractie &f)

{

cin>>f.numarator;

cin>>f.numitor;

}

void afisare3(fractie f)

{

cout<<f.numarator<<"/"<<f.numitor<<endl;

}

void produs3(fractie f1, fractie f2, fractie &fp)

{

fp.numarator=f1.numarator\*f2.numarator;

fp.numitor=f1.numitor\*f2.numitor;

}

void raport3(fractie f1, fractie f2, fractie &fr)

{

fr.numarator=f1.numarator\*f2.numitor;

fr.numitor=f1.numitor\*f2.numarator;

}

void suma3(fractie f1, fractie f2, fractie &fs)

{

fs.numarator=(f1.numarator\*f2.numitor)+(f2.numarator\*f1.numitor);

fs.numitor=f1.numitor\*f2.numitor;

}

void diferenta3(fractie f1, fractie f2, fractie &fd)

{

fd.numarator=(f1.numarator\*f2.numitor)-(f2.numarator\*f1.numitor);

fd.numitor=f1.numitor\*f2.numitor;

}

unsigned citire4(unsigned &n)

{

cin>>n;

}

float afisare4(float n)

{

cout<<n<<endl;

}

unsigned suma4(unsigned x, unsigned y)

{

return x+y;

}

int diferenta4(unsigned x, unsigned y)

{

return x-y;

}

unsigned produs4(unsigned x, unsigned y)

{

return x\*y;

}

float cat4(unsigned x, unsigned y)

{

return (float)x/y;

}

unsigned div4(unsigned x, unsigned y)

{

return x/y;

}

unsigned mod4(unsigned x, unsigned y)

{

return x%y;

}

int calcul4(unsigned x, unsigned y, unsigned z, unsigned t)

{

return (x+y)\*(z\*t-2\*x)+3\*z;

}

int fct(int x)

{

if(x<=4)

return x+1;

else

return 2\*x-1;

}

int E(int x,int y)

{

if(x<=y)

return x+y;

else

return 2\*x\*y-1;

}

int factorial5(int n)

{

int p=1,i;

for(i=n;i>=1;i--)

p=p\*i;

return p;

}

int modul5(int n)

{

if(n<0)

return -n;

else

return n;

}

int suma5(int n)

{

int i,s=0;

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

s=s+i;

return s;

}

int factorial(unsigned n)

{

int p=1,i;

for(i=n;i>=1;i--)

p=p\*i;

return p;

}

int aranjamente(int n,int kk)

{

float a,t;

t=n-kk;

a=(float)(factorial(n)/factorial(t));

return a;

}

int combinari(int n, int kk)

{

float c,t;

t=n-kk;

c=(float)(factorial(n)/(factorial(kk)\*factorial(t)));

return c;

}

void citire6(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare6(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

void citire7(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare7(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

void suma7(unsigned n, int v[100], int z[100], int s[100])

{

unsigned i;

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

s[i]=v[i]+z[i];

}

void citire8(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare8(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

int suma8(unsigned n, int v[100])

{

unsigned i,s=0;

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

s=s+v[i];

return s;

}

void citire9(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare9(unsigned n, int v[100])

{

unsigned i;

for(i=n;i>=1;i--)

cout<<v[i]<<" ";

cout<<endl;

}

void citire10(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare10(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

int produs10(unsigned n, int v[100], int z[100])

{

unsigned i,ps=0;

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

ps=ps+v[i]\*z[i];

return ps;

}

void citire11(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare11(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

void produs11(unsigned n, int v[100], int z[100], int pv[100])

{

unsigned i;

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

pv[i]=v[i]\*z[i];

}

void citire12(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare12(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

int max12(unsigned n, int v[100])

{

unsigned i,max;

max=v[1];

for(i=2;i<=n;i++)

if(v[i]>max)

max=v[i];

return max;

}

void citire13(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare13(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

int min13(unsigned n, int v[100])

{

unsigned i,min;

min=v[1];

for(i=2;i<=n;i++)

if(v[i]<min)

min=v[i];

return min;

}

void citire15(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare15(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

void sortare15(unsigned n, int v[100])

{

unsigned i,j,aux;

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

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

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

{

aux=v[i];

v[i]=v[j];

v[j]=aux;

}

}

void citire16(unsigned n, int v[100])

{

unsigned i;

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

cin>>v[i];

}

void afisare16(unsigned n, int v[100])

{

unsigned i;

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

cout<<v[i]<<" ";

cout<<endl;

}

void bule16(unsigned n, int v[100])

{

unsigned i,t=0,aux;

while(!t)

{

t=1;

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

if(v[i]<v[i+1])

{

aux=v[i];

v[i]=v[i+1];

v[i+1]=aux;

t=0;

}

}

}

void citire17(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare17(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

void citire117(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare117(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

void suma117(unsigned n, int a[20][20], int b[20][20], int s[20][20])

{

unsigned i,j;

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

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

s[i][j]=a[i][j]+b[i][j];

}

void citire18(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare18(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

void produs18(unsigned n, int a[20][20], int b[20][20], int p[20][20])

{

unsigned i,j;

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

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

p[i][j]=a[i][j]\*b[i][j];

}

void citire19(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare19(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

void produs19(unsigned n, unsigned sc, int a[20][20], int p[20][20])

{

unsigned i,j;

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

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

p[i][j]=sc\*a[i][j];

}

void citire20(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare20(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

int suma20(unsigned n, int a[20][20])

{

unsigned i,j;

int s=0;

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

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

s=s+a[i][j];

return s;

}

void citire21(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare21(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

int suma21(unsigned n, int a[20][20])

{

unsigned i,j;

int s=0;

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

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

if(i==j)

s=s+a[i][j];

return s;

}

void citire111(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare111(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

int produs111(unsigned n, int a[20][20])

{

unsigned i,j;

int p1=1;

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

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

if(i<j)

p1=p1\*a[i][j];

return p1;

}

int produs222(unsigned n, int a[20][20])

{

unsigned i,j;

int p2=1;

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

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

if(i>j)

p2=p2\*a[i][j];

return p2;

}

void citire22(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare22(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

int maximul22(unsigned n, int a[20][20])

{

unsigned i,j;

int max;

max=a[1][1];

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

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

if(a[i][j]>max)

max=a[i][j];

return max;

}

void citire23(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cin>>a[i][j];

}

void afisare23(unsigned n, int a[20][20])

{

unsigned i,j;

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

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

cout<<a[i][j]<<" ";

cout<<endl;

}

int minimul23(unsigned n, int a[20][20])

{

unsigned i,j;

int min;

min=a[1][1];

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

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

if(a[i][j]<min)

min=a[i][j];

return min;

}

float ecuatie1199(float b,float c)

{

float x;

x=-c/b;

return x;

}

float delta199(float a,float b,float c)

{

float d;

d=b\*b-4\*a\*c;

return d;

}

void caz1199(float a,float b,float c)

{

float x1,x2;

cout<<"Ecuatia are doua solutii reale diferite:"<<endl;

x1=(-b+sqrt(delta199(a,b,c)))/(2\*a);

x2=(-b-sqrt(delta199(a,b,c)))/(2\*a);

cout<<"x1="<<x1<<endl;

cout<<"x2="<<x2;

}

void caz2199(float a,float b,float c)

{

float x1,x2;

cout<<"Ecuatia are doua solutii reale identice:"<<endl;

x1=-b/(2\*a);

x2=-b/(2\*a);

cout<<"x1="<<x1<<endl;

cout<<"x2="<<x2;

}

void caz3199(float a,float b,float c)

{

float d;

d=delta199(a,b,c);

cout<<"Ecuatia are doua solutii complexe diferite:"<<endl;

cout<<"x1=("<<-b<<"+"<<sqrt(-d)<<"i)"<<"/"<<2\*a<<endl;

cout<<"x2=("<<-b<<"-"<<sqrt(-d)<<"i)"<<"/"<<2\*a<<endl;

}

float a,b,c;

float ecuatie1198()

{

float x;

x=-c/b;

return x;

}

float delta198()

{

float d;

d=b\*b-4\*a\*c;

return d;

}

void caz1198()

{

float x1,x2;

cout<<"Ecuatia are doua solutii reale diferite:"<<endl;

x1=(-b+sqrt(delta198()))/(2\*a);

x2=(-b-sqrt(delta198()))/(2\*a);

cout<<"x1="<<x1<<endl;

cout<<"x2="<<x2;

}

void caz2198()

{

float x1,x2;

cout<<"Ecuatia are doua solutii reale identice:"<<endl;

x1=-b/(2\*a);

x2=-b/(2\*a);

cout<<"x1="<<x1<<endl;

cout<<"x2="<<x2;

}

void caz3198()

{

float d;

d=delta198();

cout<<"Ecuatia are doua solutii complexe diferite:"<<endl;

cout<<"x1=("<<-b<<"+"<<sqrt(-d)<<"i)"<<"/"<<2\*a<<endl;

cout<<"x2=("<<-b<<"-"<<sqrt(-d)<<"i)"<<"/"<<2\*a<<endl;

}

void citire100(int v[12])

{

unsigned i;

for(i=1;i<=12;i++)

cin>>v[i];

}

int max100(int v[100])

{

unsigned i,max;

max=v[1];

for(i=2;i<=12;i++)

if(v[i]>max)

max=v[i];

return max;

}

int min100(int v[100])

{

unsigned i,min;

min=v[1];

for(i=2;i<=12;i++)

if(v[i]<min)

min=v[i];

return min;

}

int media100(int v[100])

{

unsigned i,m,s=0;

for(i=1;i<=12;i++)

s=s+v[i];

m=s/12;

return m;

}

void sortare100(int v[100])

{

unsigned i,j,aux;

for(i=1;i<=12-1;i++)

for(j=i+1;j<=12;j++)

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

{

aux=v[i];

v[i]=v[j];

v[j]=aux;

}

}

void bule100(int v[100])

{

unsigned i,t=0,aux;

while(!t)

{

t=1;

for(i=1;i<=12-1;i++)

if(v[i]<v[i+1])

{

aux=v[i];

v[i]=v[i+1];

v[i+1]=aux;

t=0;

}

}

}

fstream f11("vector.txt",ios::in), f22("suma.txt",ios::out);

void citire24(unsigned n, unsigned a[100])

{

int i;

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

f11>>a[i];

}

unsigned palindrom24(int n)

{

int inv=0,m;

m=n;

while(n!=0)

{

inv=inv\*10+n%10;

n=n/10;

}

if(inv==m)

return 1;

else

return 0;

}

fstream f111("vector.txt",ios::in);

void citire25(unsigned n, unsigned a[100])

{

int i;

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

f111>>a[i];

}

int divizor25(int n)

{

int i,m=0;

for(i=2;i<=n/2;i++)

{

if(n%i==0)

m++;

}

return m;

}

fstream f1111("vector.txt",ios::in), f2222("solutie.txt",ios::out);

void citire26(unsigned n, unsigned a[100])

{

int i;

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

f1111>>a[i];

}

int perfect26(int n)

{

int i,m=1,t;

t=n;

for(i=2;i<=n/2;i++)

{

if(n%i==0)

m=m+i;

}

if(t==m)

return 1;

else

return 0;

}

fstream f11111("vector.txt",ios::in);

void citire27(int n, int a[100])

{

int i;

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

f11111>>a[i];

}

void afisare27(int n, int a[100])

{

unsigned i;

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

cout<<a[i]<<" ";

}

int min27(int n, int a[100])

{

int min,i;

min=a[1];

for(i=2;i<=n;i++)

if(a[i]<min)

min=a[i];

return min;

}

void inte(int &a, int &b)

{

int aux;

aux=a;

a=b;

b=aux;

}

int max31(unsigned n, int a)

{

int max=0,i;

max=a;

for(i=2;i<=n;i++)

{

cin>>a;

if(a>max)

max=a;

}

return max;

}

int min32(int a)

{

int min=0;

min=a;

while(a!=0)

{

if(a<min)

min=a;

cin>>a;

}

return min;

}

void extragerea33(int n, int c)

{

while(n!=0)

{

c=n%10;

cout<<"c="<<c<<endl;

n=n/10;

}

}

int compunerea34(int c)

{

int nr=0;

while(c>=0&&c<=9)

{

nr=nr\*10+c;

cin>>c;

}

return nr;

}

int inversul35(int n)

{

int inv=0;

while(n!=0)

{

inv=inv\*10+n%10;

n=n/10;

}

return inv;

}

int cmmdc36(int a,int b)

{

int r;

while(b!=0)

{

r=a%b;

a=b;

b=r;

}

return a;

}

int cmmdc37(int a,int b)

{

while(a!=b)

{

if(a>b)

a=a-b;

else

b=b-a;

}

return a;

}

int prim41(int n)

{

int x=1,i=2;

while(i<=sqrt(n)&&x==1)

{

if(n%i==0)

x=0;

else

i=i+1;

}

if(x==1)

return 1;

else

return 2;

}

void divizor42(int n)

{

int i;

cout<<1<<" ";

for(i=2;i<=n/2;i++)

{

if(n%i==0)

cout<<i<<" ";

}

cout<<n;

}

void divizor43(int n)

{

int i;

cout<<1<<" "<<n<<" ";

for(i=2;i<=sqrt(n);i++)

{

if(n%i==0)

cout<<i<<" "<<n/i<<" ";

}

}

void recurente44(unsigned n)

{

unsigned i,anteprec,prec,fibo;

anteprec=1;

prec=1;

cout<<anteprec<<" "<<prec<<" ";

for(i=3;i<=n;i++)

{

fibo=anteprec+prec;

cout<<fibo<<" ";

anteprec=prec;

prec=fibo;

}

}

int main()

{

int cazul, prog, prog1, prog2, prog3,ab,lectii,subtitlu;

cout<<"Introduceti numarul cazului:"<<endl;

cout<<"1.Pentru teoria de la subprograme;"<<endl;

cout<<"2.Pentru programe implementate cu ajutorul subprogramelor."<<endl;

cin>>cazul;

while(cazul!=3)

{

switch(cazul)

{

case 1:

{

cout<<"Meniul pentru teoria de la subprograme: "<<endl;

cout<<"1.Notiuni introductive"<<endl;

cout<<"2.Clasificarea subprogramelor"<<endl;

cout<<"3.Reguli pentru construirea subprogramelor C++"<<endl;

cout<<"4.Utilizarea stivei de catre subprograme"<<endl;

cout<<"5.Tablourile de memorie si subprogramele"<<endl;

cout<<"6.Clasificarea variabilelor de memorie"<<endl;

cout<<"7.Pentru intoarcerea la meniul principal"<<endl;

cout<<"Introduceti cazul dorit: ";cin>>lectii;

cout<<endl;

while(lectii!=7)

{

switch(lectii)

{

case 1:

{

cout<<endl;

cout<<" Blocul este unitatea de baza a oricarui program C++."<<endl;

cout<<" Acesta este incapsulat intr-o instructiune compusa si este delimitat de {...}."<<endl;

cout<<" Orice program contine un bloc principal, numit functie radacina( int main() )."<<endl;

cout<<" Un bloc este compus din doua categorii de instructiuni( instructiuni "<<endl;

cout<<"declarative si intructiuni procedural sau executabile ). Blocul are forma:"<<endl;

cout<<" \_\_\_\_\_\_\_\_\_"<<endl;

cout<<"|antet |"<<endl;

cout<<"|{corp} <-|----- instructiuni declarative si procedurale"<<endl;

cout<<"|\_\_\_\_\_\_\_\_\_|"<<endl;

cout<<endl;

cout<<" Studiu de caz:"<<endl;

cout<<"int main() → antet"<<endl;

cout<<"{"<<endl;

cout<<" \* int a,b; |--->Instructiuni declarative"<<endl;

cout<<" \* float ma; |--->Instructiuni declarative"<<endl;

cout<<" \* cout<<'a='; \*"<<endl;

cout<<" \* cin>>a; \*"<<endl;

cout<<"corp \* cout<<'b='; \*"<<endl;

cout<<" \* cin>>b; \* instructiuni procedurale"<<endl;

cout<<" \* ma=(float)(a+b)/2; \*"<<endl;

cout<<" \* cout<<'Media='<<ma; \*"<<endl;

cout<<" \* return 0; \*"<<endl;

cout<<"}"<<endl;

cout<<endl;

cout<<"Definitie!"<<endl;

cout<<" Subprogramul este o secventa de instructiuni ce rezolva o anumita sarcina si"<<endl;

cout<<"care poate fi scrisa separat de blocul radacaina si lansata in executie"<<endl;

cout<<"din interiorul oricarui bloc, ori de cate ori este nevoie."<<endl;

cout<<" Utilitatea subprogramelor intervine atunci cand o problema poate fi descompusain mai multe subprobleme( numite module ) sau cand o secventa de instructiuni serepeta in acelasi program."<<endl;

cout<<" Intr-un program exista doua categorii de module:"<<endl;

cout<<"- modulul apelant: este cel care pentru rezolvarea proriilor sarcini apeleaza laalt modul( modulul apelat );"<<endl;

cout<<"- modulul apelat: este cel care rezolva sarcinile promite de la modulul apelant."<<endl;

cout<<endl;

cout<<"Modulul apelant(P) Modulul apelat(SP)"<<endl;

cout<<" ---------- ------------------"<<endl;

cout<<"| -inst. 1 | | - |"<<endl;

cout<<"| -inst. 2 | | - |"<<endl;

cout<<"| . | | - |"<<endl;

cout<<"| . | | - |"<<endl;

cout<<"| . | \*>| - |"<<endl;

cout<<"| -inst. i | \* | - |"<<endl;

cout<<"| . | \* | - |"<<endl;

cout<<"| . | \* | - |"<<endl;

cout<<"| . | Trasfera controlul \* | - |"<<endl;

cout<<"| apel--|----------------------\* | - |"<<endl;

cout<<"| | Revenirea controlului \* | - |"<<endl;

cout<<"| -inst. j | \* | - |"<<endl;

cout<<"| . | \* | - |"<<endl;

cout<<"| . | \* | - |"<<endl;

cout<<"| . | \*<| - |"<<endl;

cout<<"| -inst. n | | - |"<<endl;

cout<<" ---------- ------------------"<<endl;

cout<<endl;

cout<<" Comunicarea intre module se realizeaza prin intermediul parametrilor."<<endl;

cout<<" Parametrii sunt de trei tipuri:"<<endl;

cout<<"- parametrii de intrare( sunt transmisi de catre programul principal"<<endl;

cout<<"subprogramului, pot fi prelucrati doar in intermediul subprogramului SP)."<<endl;

cout<<"P -> SP"<<endl;

cout<<"- parametrii de iesire( sunt transmisi de catre subprogram, programului"<<endl;

cout<<"principal; el reprezinta de fapt rezultatele, pot fi prelucrati doar de catre programul principal P). SP -> P"<<endl;

cout<<"- parametrii de intrare-iesire, au dublu rol si pot fi prelucrati atat in"<<endl;

cout<<"interiorul programului P cat si in interiorul subprogramului SP. P <-> SP"<<endl;

cout<<endl;

cout<<" Observatie!"<<endl;

cout<<" In limbajul C++ subprogramele se mai numesc functii."<<endl;

cout<<" Elementele unui subprogram sunt:"<<endl;

cout<<"- definitia subprogramului, specifica ceea ce realizeaza subprogramul"<<endl;

cout<<"- apelul subprogramului, reprezinta lansarea in executie a subprogramului"<<endl;

cout<<"- prototipul subprogramului, ofera compilatorului informatii despre subprogram"<<endl;

cout<<endl;

cout<<" Studiu de caz:"<<endl;

cout<<"#include<iostream>"<<endl;

cout<<"using namespace std;"<<endl;

cout<<"void scrie(); --->prototipul"<<endl;

cout<<"int main()"<<endl;

cout<<"{"<<endl;

cout<<"scrie(); --->apel"<<endl;

cout<<"return 0;"<<endl;

cout<<"}"<<endl;

cout<<"void scrie() --->antet \*"<<endl;

cout<<"{ \* definitie"<<endl;

cout<<"cout<<'Hello C++!'; --->corp \*"<<endl;

cout<<"}"<<endl;

cout<<endl;

cout<<" Observatie!"<<endl;

cout<<" In cazul in care definitia subprogramului este scrisa inaintea de blocul in"<<endl;

cout<<"care acesta este apelata, prototipul poate sa lipseasca. Deoarece antetul din"<<endl;

cout<<"definitia subprogramului are rol declarative."<<endl;

cout<<endl;

cout<<"#include<iostream>"<<endl;

cout<<"using namespace std;"<<endl;

cout<<"void scrie()"<<endl;

cout<<"{"<<endl;

cout<<"cout<<'Hello C++!';"<<endl;

cout<<"}"<<endl;

cout<<"int main()"<<endl;

cout<<"{"<<endl;

cout<<"scrie();"<<endl;

cout<<"return 0;"<<endl;

cout<<"}"<<endl;

}break;

case 2:

{

cout<<endl;

cout<<" Exista 2 criterii de clasificare:"<<endl;

cout<<"a) Dupa autor:"<<endl;

cout<<"1) Subprograme standard( de sistem )"<<endl;

cout<<"- sunt cele predefinite in limbajul de programare si au fost create chiar de"<<endl;

cout<<"autorii limbajului"<<endl;

cout<<"Ex: sqrt(x), pow(x,n), strlen(s), strcpy(d,s)"<<endl;

cout<<"- functiile utilizate in prelucrarea sirurilor de caractere, functii de citire"<<endl;

cout<<"si afisare, etc;"<<endl;

cout<<endl;

cout<<" Observatie!"<<endl;

cout<<" Aceste functii sunt definite in fisierele header care trebuie incluse in"<<endl;

cout<<"program utilizand directive de preprocesare."<<endl;

cout<<endl;

cout<<"#include<fisier\_header>"<<endl;

cout<<endl;

cout<<"2)Subprograme utilizator( nestandard )"<<endl;

cout<<"- sunt cele create de catre programatori"<<endl;

cout<<"Ex: scrie()"<<endl;

cout<<endl;

cout<<"b) Dupa modalitatea de apelare:"<<endl;

cout<<"1) Subprograme procedurale"<<endl;

cout<<"- care sunt apelate utiliznad sintaxa:"<<endl;

cout<<endl;

cout<<"nume\_subprogram(lista\_parametrii)"<<endl;

cout<<endl;

cout<<" Observatii!"<<endl;

cout<<" 1.Functiile procedural au tipul void ceea ce inseamna ca nu returneaza nimic"<<endl;

cout<<"prin numle lor. Ele pot returna rezultate prin intermediul parametrilor."<<endl;

cout<<" 2.Lista de parametrii poate fi vida."<<endl;

cout<<" 3.Este posibil ca functiile procedural sa nu returneze rezultate nici prin"<<endl;

cout<<"intermediul parametrilor."<<endl;

cout<<"2) Subprogramele operand"<<endl;

cout<<"- sunt apelate astfel:"<<endl;

cout<<"- utilizand o instructiune de afisare, de exemplu:"<<endl;

cout<<"cout<<nume\_subprogram(lista\_parametrii);"<<endl;

cout<<"- intr-o operatie de atribuire, de exemplu:"<<endl;

cout<<"int n=strlen(s);"<<endl;

cout<<"- intr-o expresie, de exemplu:"<<endl;

cout<<"s=a+nume\_subprogram(lista\_parametrii);"<<endl;

cout<<"s=a+pow(x,n);"<<endl;

cout<<endl;

cout<<" Observatie!"<<endl;

cout<<" Subprogramele operand returneaza un rezultat prin chiar numele lor si pot"<<endl;

cout<<"returna si alte rezultate prin intermediul parametrilor. Ele au un tip de date"<<endl;cout<<"elementare( int, float, unsigned ), iar rezultatul returnat de ele va fi de acest tip."<<endl;

cout<<endl;

cout<<" Sintaxele definitiilor subprgramelor procedural si operand:"<<endl;

cout<<"\* Subprogramele procedurale:"<<endl;

cout<<endl;

cout<<"void nume\_subprogram(lista\_parametrii)"<<endl;

cout<<"{"<<endl;

cout<<" \*instructiuni declarative"<<endl;

cout<<"corp \*"<<endl;

cout<<" \*instructiuni procedurale"<<endl;

cout<<"}"<<endl;

cout<<endl;

cout<<"\* Subprograme operand:"<<endl;

cout<<endl;

cout<<"tip\_data nume\_subprogram(lista\_parametrii)"<<endl;

cout<<"{"<<endl;

cout<<" \*instructiuni declarative"<<endl;

cout<<"corp \* instructiuni procedurale"<<endl;

cout<<" \*return expresie;"<<endl;

cout<<"}"<<endl;

cout<<endl;

}break;

case 3:

{

cout<<endl;

cout<<" Definitia subprogramului"<<endl;

cout<<"Definitia unui subprogram este format din antetul si corpul subprogramului:"<<endl;

cout<<endl;

cout<<"<antetul subprogramului>"<<endl;

cout<<"{"<<endl;

cout<<"<declaratii proprii subprogramului>"<<endl;

cout<<"<instructiuni>"<<endl;

cout<<"[return<expresie>;]"<<endl;

cout<<"}"<<endl;

cout<<"a) Antetul subprogramului este o linie de recunoastere a subprogramului, in care";

cout<<"i se atribuie un nume. El specifica inceputul subprogramului."<<endl;

cout<<"b) Corpul subprogramului, la fel ca orice bloc C++, este incapsulat intr-o"<<endl;

cout<<"instructiune compusa, delimitate de accolade si este format din doua parti:"<<endl;

cout<<"- partea declarativa contine definitii de elemante folosite numai in interiorul subprogramului: tipuri de date, constant si variabile de memorie. Nu se pot"<<endl;

cout<<"definii si alte subprograme(nu este variabila tehnica de imbricare a"<<endl;

cout<<"subprogramelor existent in alte limbaje de programare)."<<endl;

cout<<"- Partea executabila contine instructiunile prin care sunt descrise actiunile"<<endl;cout<<"realizate de subprogram."<<endl;

cout<<endl;

cout<<"Introduceti un numar pentru informatii despre:"<<endl;

cout<<"1.Antetul subprogramului"<<endl;

cout<<"2.Corpul subprogramului"<<endl;

cout<<"3.Prototipul subprogramului"<<endl;

cout<<"4.Activarea (apelul) subprogramului"<<endl;

cout<<"5.Parametrii de comunicare"<<endl;

cout<<"6.Transferul parametrilor intre subprograme"<<endl;

cout<<"7.Revenire la meniul anterior"<<endl;

cout<<"Introduceti cazul dorit"<<endl;

cin>>subtitlu;

while(subtitlu!=7)

{

switch(subtitlu)

{

case 1:

{

cout<<" Subprogramul trebuie sa aiba un antet prin care se precizeaza interfata dintre"<<endl;

cout<<"programul apelat si subprogram. El contine 3 categorii de informatii:"<<endl;

cout<<"\* Tipul rezultatului pentru functiile operand se precizeaza tipul furnizat de"<<endl;

cout<<"subprogram prin chiar numele sau. Pentru functiile procedural, tipul"<<endl;

cout<<"rezultatului este void( nu intoarce nici un rezultat prin numele sau;"<<endl;

cout<<"rezultatele vor fi intoarse prin parametrii subprogramului). Daca nu se"<<endl;

cout<<"precizeaza tipul rezultatului, compilatorul va considera ca acesta este implicit";

cout<<"de tip int."<<endl;

cout<<"\* Numele subprogramului este un identificator unic, care se atribuie"<<endl;

cout<<"subprogramului. Numele trebuie sa respecte aceleasi reguli ca orice"<<endl;

cout<<"identificator C++."<<endl;

cout<<"\* Parametrii folositi pentru comunicare; pentru fiecare parametru se precizeaza numele si tipul."<<endl;

cout<<endl;

cout<<" Antetul unui subprogram este de forma:"<<endl;

cout<<"tip rezultat nume\_subprogram(lista\_parametrii)"<<endl;

cout<<" Lista de parametrii este de forma:"<<endl;

cout<<"(tip 1 p1, tip 2 p2, tip 3 p3,…, tip n pn)"<<endl;

cout<<" | | |"<<endl;

cout<<" | | |--> separator"<<endl;

cout<<" | |--> identificatorul parametrului"<<endl;

cout<<" |--> tipul parametrului"<<endl;

cout<<endl;

cout<<"Exemplu 1: float alfa(int a, int b, float c)"<<endl;

cout<<" Acesta este antetul unei functii operand care furnizeaza un rezultat de tip"<<endl;

cout<<"float. Numele functiei este alfa, iar parametrii folositi pentru comunicare sunt"<<endl;

cout<<"a si b de tip int si c de tip float."<<endl;

cout<<"Exemplul 2: void beta(int a, float b, float c, char d)"<<endl;

cout<<"Exemplul 3: void gama()"<<endl;

cout<<endl;

cout<<" Observatii!"<<endl;

cout<<"1) Separarea parametrilor in lista se face prin ','.Daca exista mai multi"<<endl;

cout<<"parametrii de acelasi tip, ei nu pot fi grupati ca la declararea tipului"<<endl;

cout<<"variabilelor de memorie. Pentru fiecare parametru trebuie precizat tipul."<<endl;

cout<<"2) Tipul parametrilor poate fi:"<<endl;

cout<<"\* Orice tip standard al sistemului folosit pentru datele elementare- intreg"<<endl;

cout<<"(int, unsigned, long), real(double, float, long double) sau caracter(char sau"<<endl;

cout<<"unsigned char), -tipul pointer sau orice tip de structura de data( vector,"<<endl;

cout<<"matrice, sir de caractere sau inregistrare)"<<endl;

cout<<"\* Orice tip definit de utilizator inainte de a define subprogramul."<<endl;

cout<<endl;

cout<<"3)Pentru rezultatul functiei nu se poate folosi tipul tablou de memorie."<<endl;

cout<<"Exemplu: float a[10] tablou(int v, unsigned n)"<<endl;

cout<<" Acest antet de subprogram va produce eroare deoarece tipul functiei este"<<endl;

cout<<"tablou de memorie."<<endl;

cout<<"4) Numele subprogramului poate fi folosit in trei locuri distincte:"<<endl;

cout<<"\* In prototipul subprogramului, unde are un rol declarative;"<<endl;

cout<<"\* In antetul subprogramului, unde are un rol de definitie, dar si declarative;"<<endl;

cout<<"\* In apelarea subprogramului, unde are rol de activare."<<endl;

}break;

case 2:{

cout<<"Corpul subprogramului este un bloc care contine atat instructiuni declarative,"<<endl;

cout<<"cat si instructiuni imperative. Variabilele de memorie declarate in corpul"<<endl;

cout<<"subprogramului se numesc variabile locale. In cazul unei functii operand, ultima";

cout<<"instructiune din corpul subprogramului trebuie sa fie instructiunea return, care";

cout<<"are sintaxa:"<<endl;

cout<<endl;

cout<<"return<expresie>;"<<endl;

cout<<endl;

cout<<"Valoarea obtinuta prin evaluarea expresiei <expresie> va fi atribuita functiei"<<endl;

cout<<"operand( va fi valoarea returnata prin numele functiei). Rezultatul expresiei"<<endl;

cout<<"trebuie sa fie de acelasi tip cu tipul functiei sau de un tip care poate fi"<<endl;

cout<<"convertit implicit in tipul functiei."<<endl;

cout<<"Cand compilatorul C++ intalneste intr-un subprogram instructiunea return,"<<endl;

cout<<"termina executia subprogramului si reda controloul modulului appellant. Prin"<<endl;

cout<<"urmare, daca veti scrie in subprogram dupa instructiunea return alte"<<endl;

cout<<"instructiuni, ele nu vor fi executate."<<endl;

}break;

case 3:

{

cout<<" Este o linie de program, aflata inaintea modulului care apeleaza subprogramul,";

cout<<"prin care se comunica compilatorului informatii despre subprogram( se declara"<<endl;

cout<<"subprogramul). Prin declararea programului, compilatorul primeste informatii"<<endl;

cout<<"despre modulul in care se poate apela subprogramul si poate face verificari la"<<endl;

cout<<"apelurile de subprogram in ceea ce priveste tipul parametrilor folositi pentru"<<endl;

cout<<"comunicare si amodulului in care poate face conversia acestor parametrii."<<endl;

cout<<" Un subprogram, pentru a putea fi folosit, trebuie declarat. Pentru declararea"<<endl;

cout<<"lui se foloseste prototipul. El contine 3 categorii de informatii, la fel ca si"<<endl;

cout<<"antetul subprogramului: tipul rezultatului, numele subprogramului si tipul"<<endl;

cout<<"parametrilor folositi pentru comunicare. Pentru fiecare parametru din antetul"<<endl;

cout<<"subprogramului, se pot preciza numai tipul, nu si numele lui."<<endl;

cout<<" Prototipul unui subprogram este de forma:"<<endl;

cout<<"tip rezultat nume\_subprogram(lista \_tipuri\_parametrii)"<<endl;

cout<<" Lista tipurilor parametrilor este de forma:"<<endl;

cout<<"tip 1, tip 2, tip 3, ..., tip n"<<endl;

cout<<" Observatii:"<<endl;

cout<<"1) Separarea tipurilor de parametrii in lista se face prin caracterul ','. Lista";

cout<<"trebuie sa contina atatea tipuri de parametrii cati parametrii au fost"<<endl;

cout<<"definite in antetul subprogramului. In lista se precizeaza tipul de data la care";

cout<<"se refera, in aceeasi ordine in care au fost scrisi parametrii la definirea lor"<<endl;

cout<<"in antet."<<endl;

cout<<"2) Spre deosebire de antetul subprogramului , prototipul se termina cu"<<endl;

cout<<"caracterul ';'."<<endl;

cout<<" Pentru functiile al caror antet a fost precizat anterior,prototipurile vor fi:"<<endl;

cout<<"Exemplul 1: float alfa(int , int, float);"<<endl;

cout<<"Exemplul 2: void beta(int , float, float, char);"<<endl;

cout<<"Exemplul 3: void game();"<<endl;

cout<<endl;

}break;

case 4:

{

cout<<" Subprogramul trebuie sa fie cunoscut atunci cand se cere prin apel activarea"<<endl;

cout<<"lui."<<endl;

cout<<"\* Daca subprogramul este standard, trebuie inclus fisierul care contine"<<endl;

cout<<"prototipul subprogramului in fisierul sursa."<<endl;

cout<<"\* Daca subprogramul este utilizator, trebuie declarat fie prin folosirea"<<endl;

cout<<"prototipului,fie prin definirea lui inaintea apelului."<<endl;

cout<<" In functie de modul in care a fost definit, subprogramul se activeaza fie"<<endl;

cout<<"printr-o instructiune procedural, fie ca operand intr-o expresie."<<endl;

cout<<" Pentru functiile ale caror antet a fost precizat anterior, activarea se poate face astfel:"<<endl;

cout<<"Exemplul 1: w=alfa(x,y,z);"<<endl;

cout<<"Exemplul 2: int x; float y,z; char w;"<<endl;

cout<<"beta(x,y,z,w);"<<endl;

cout<<"Exemplul 3: gama();"<<endl;

cout<<endl;

cout<<" Orice subprogram trebuie declarat si definit. Declararea unui subprogram este necesara pentru ca el sa fie cunoscut de subprogram care il apeleaza. Declararea";

cout<<"lui poate fi facuta fie prin prototip, fie prin definitia lui (antetul"<<endl;

cout<<"impreuna cu corpul subprogramului). Pentru a declara subprogramul, fie scrieti"<<endl;

cout<<"prototipul inaintea subprogramului care il apeleaza, putand scrie apoi definitia"<<endl;

cout<<"lui oriunde in program, fie definiti subprogramul inaintea subprogramului care"<<endl;

cout<<"il apeleaza."<<endl;

cout<<endl;

cout<<"#include<iostream>"<<endl;

cout<<"void scrie() <-- antetul subprogramului \*"<<endl;

cout<<"{ \* definitia subprogramului"<<endl;

cout<<"cout<<'Subprogram';} <-- corpul subprogramului \*"<<endl;

cout<<"void main()"<<endl;

cout<<"{"<<endl;

cout<<"scrie(); <-- apelul subprogramului"<<endl;

cout<<"}"<<endl;

cout<<" Asadar:"<<endl;

cout<<"\* Prototipul subprogramului declara subprogramul"<<endl;

cout<<"\* Apelul subprogramului executa subprogramul"<<endl;

cout<<"\* Antetul subprogramului specifica numele subprogramului si tipul argumentelor si al valorilor returnate, iar corpul subprogramului il defineste, adica"<<endl;

cout<<"specifica ceea ce trebuie sa realizeze subprogramul."<<endl;

cout<<endl;

}break;

case 5:

{

cout<<" Daca pentru comunicarea intre subprogram si bloc apelant se folosesc"<<endl;

cout<<"parametrii,acestia vor fi scrisi dupa numele subprogramului, intre paranteze"<<endl;

cout<<"rotunde, astfel:"<<endl;

cout<<"\* In antet, pentru fiecare parametru se precizeaza denumirea simbolica folosita";

cout<<"in interiorul subprogramului. Acesti parametrii se numesc parametrii formali."<<endl;

cout<<"In prototip, pentru fiecare parametru se precizeaza tipul de data la care se"<<endl;

cout<<"refera, in aceeasi ordine in care au fost scrisi la definirea lor in antet."<<endl;

cout<<"\* La activarea subprogramului, parametrilor de comunicare li se vor atribui"<<endl;

cout<<"valori concrete cu care se va executa subprogramul la acest apel. Aceste valori"<<endl;

cout<<"vor fi communicate la apelul subprogramului, dupa numele subprogramului, intre"<<endl;

cout<<"paranteze rotunde, in aceeas ordine in care au fost scrisi la definirea lor in"<<endl;

cout<<"antet. Comunicarea trebuie sa respecte aceeasi succesiune, tipuri de date si"<<endl;

cout<<"numar de parametrii ca si in lista de parametrii formali, deoarece atribuirea"<<endl;

cout<<"valorilor se face respectand regula de corespondenta. Acesti parametrii"<<endl;

cout<<"se numesc parametrii actuali."<<endl;

cout<<" Observatii!"<<endl;

cout<<"1) Numele parametrilor actuali poate fi diferit de numele parametrilor formali.";

cout<<"2) Tipul parametrilor formali poate sa fie diferit de tipul parametrilor"<<endl;

cout<<"actuali numai cand parametrii actuali pot fi convertiti la tipul celor formali."<<endl;

cout<<endl;

}break;

case 6:

{

cout<<" Transferul parametrilor reprezinta tehnica utilizata pentru comunicarea"<<endl;

cout<<"datelor intre module."<<endl;

cout<<" Exista urmatoarele metode de transfer:"<<endl;

cout<<"a) Transferul prin valoare:"<<endl;

cout<<"\* In momentul apelarii subprogramului valoarea parametrilor este incarcata in"<<endl;

cout<<"stiva sistemului"<<endl;

cout<<"\* Acesti parametrii se numesc parametrii valoare."<<endl;

cout<<"\* Valoarea lor din stiva este accesibila doar modulului apelat pe parcursul"<<endl;

cout<<"executiei acestuia."<<endl;

cout<<"\* Dupa incheierea executiei modulului apelat valoarea din stiva se pierde."<<endl;

cout<<"\* Aceasta metoda de transfer poate fi utilizata doar pentru parametriii de"<<endl;

cout<<"intrare."<<endl;

cout<<"Exemplu: void suma(int a, int b)"<<endl;

cout<<"{...}"<<endl;

cout<<"\* Pentru a utilize transferul prin valoare pentru parametrii de iesire sau"<<endl;

cout<<"pentru cei de intrare-iesire se utilizeaza variabilele de tip pointer."<<endl;

cout<<"Exemplu: void suma(int a, int b, int \*s)"<<endl;

cout<<"{...}"<<endl;

cout<<"suma (a,b,&s);"<<endl;

cout<<"\* Astfel pentru transmiterea rezultatelor din subprogram catre programul"<<endl;

cout<<"principal se utilizeaza transferul prin valoare utilizand pointer."<<endl;

cout<<"b) Transferul prin referinta( adresa )"<<endl;

cout<<"\* Parametrii transferati prin aceasta metoda se numesc parametrii variabila."<<endl;

cout<<"\* Parametrii pastreaza adresa din stiva astfel ca valoarea acestora este"<<endl;

cout<<"reflectata si dupa incheierea executiei subprogramului si in programul"<<endl;

cout<<"principal; aceasta tehnica se foloseste pentru transmiterea parametrilor"<<endl;

cout<<"(iesire,intrare-iesire)"<<endl;

cout<<"Exemplu: void suma( int a, int b, int &s)"<<endl;

cout<<"{...}"<<endl;

cout<<"suma(a,b,s);"<<endl;

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5,6 sau 7"<<endl;

}

cout<<endl;

cout<<"Introduceti un numar pentru informatii despre:"<<endl;

cout<<"1.Antetul subprogramului"<<endl;

cout<<"2.Corpul subprogramului"<<endl;

cout<<"3.Prototipul subprogramului"<<endl;

cout<<"4.Activarea (apelul) subprogramului"<<endl;

cout<<"5.Parametrii de comunicare"<<endl;

cout<<"6.Transferul parametrilor intre subprograme"<<endl;

cout<<"7.Revenire la meniul anterior"<<endl;

cout<<"Introduceti cazul dorit:"<<endl;

cin>>subtitlu;

}

}break;

case 4:

{

cout<<" In memoria interna exista o zona anumita stiva sistemului( stack ) in care se pastreaza temporar informatii despre subprogram. Aceste informatii formeaza"<<endl;

cout<<"instant subprogramului care este formata din:"<<endl;

cout<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" \*Instanta subprogramului\*"<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" | "<<endl;

cout<<" | "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*Adresa de revenire\*<->\*Contractul de apel\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" | "<<endl;

cout<<" | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" \*Parametrii de comunicare\*<->\*Parametrii locali\*"<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<endl;

cout<<" Adresa de revenire reprezinta adresa instructiunii din modulul apelant care"<<endl;

cout<<"urmeaza imediat dupa apelul subprogramului."<<endl;

cout<<endl;

cout<<" Etapele execuatate la apelul subprogramului sunt:"<<endl;

cout<<"a) Se preda controlul modulului apelat"<<endl;

cout<<"b) Se pregateste stiva sistemului astfel:"<<endl;

cout<<"\* Se salveaza/incarca in stiva adresa de revenire"<<endl;

cout<<"\* Se incarca valorile parametrilor( de la dreapta la stanga )"<<endl;

cout<<"\* Se rezerva spatiu pentru variabilele locale."<<endl;

cout<<"c) Se executa instructiunile din corpul modulului apelat."<<endl;

cout<<endl;

cout<<"Exemplu: alfa(x,y,z)"<<endl;

cout<<" Stiva: "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* Variabile locale \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* x \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* y \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* z \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*Adresa de revenire\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<endl;

cout<<" Etapele executate la incheierea executarii programului sunt:"<<endl;

cout<<"a) Se intrerupe executia modulului apelat"<<endl;

cout<<"b) Se elibereaza stiva sistemului:"<<endl;

cout<<"\* Se elibereaza spatial ocupat de parametrii"<<endl;

cout<<"\* Se extrage adresa de revenire"<<endl;

cout<<"c) Se reda controlul modulului apelant si se continua executia acestuia cu"<<endl;

cout<<"instructiunea de la adresa de revenire."<<endl;

}break;

case 5:

{

cout<<"Citirea si afisarea vectorului cu ajutorul subprogramelor:"<<endl;

cout<<endl;

cout<<"void citire( unsigned n, int v[100])"<<endl;

cout<<"{unsigned i;"<<endl;

cout<<"for(i=1;i<=n;i++)"<<endl;

cout<<"cin>>v[i];"<<endl;

cout<<"}"<<endl;

cout<<"void afisare( unsigned n, int v[100])"<<endl;

cout<<"{unsigned i;"<<endl;

cout<<"for(i=1;i<=n;i++)"<<endl;

cout<<"cout<<v[i]<<' ';"<<endl;

cout<<"}"<<endl;

cout<<"int main()"<<endl;

cout<<"{int x[100],y[100]; unsigned n;"<<endl;

cout<<"cin>>n; citire(n,x); citire(n,y); afisare(n,x); afisare(n,y);"<<endl;

cout<<"return 0;"<<endl;

cout<<"}"<<endl;

cout<<endl;

cout<<"Citirea si afisarea unei matrice cu ajutorul subprogramelor:"<<endl;

cout<<endl;

cout<<"void citire( unsigned n, unsigned m, int a[20][30])"<<endl;

cout<<"{unsigned i,j;"<<endl;

cout<<"for(i=1;i<=n;i++)"<<endl;

cout<<"for(j=1;j<=m;j++)"<<endl;

cout<<"cin>>a[i][j];"<<endl;

cout<<"}"<<endl;

cout<<"void afisare( unsigned n, unsigned m, int a[20][30])"<<endl;

cout<<"{unsigned i,j;"<<endl;

cout<<"for(i=1;i<=n;i++)"<<endl;

cout<<"{ for(j=1;j<=m;j++)"<<endl;

cout<<"cout<<a[i][j]<<' ';"<<endl;

cout<<"cout<<endl;}"<<endl;

cout<<"}"<<endl;

cout<<"int main()"<<endl;

cout<<"{int x[20][30],y[20][30]; unsigned n,m;"<<endl;

cout<<"cin>>n; cin>>m; citire(n,m,x); citire(n,m,y); afisare(n,m,x); afisare(n,m,y);"<<endl;

cout<<"return 0;"<<endl;

cout<<"}"<<endl;

cout<<endl;

}break;

case 6:

{

cout<<" Exista in memoria interna urmatoarele zone in care se poate rezerva spatiu"<<endl;

cout<<"pentru variabile:"<<endl;

cout<<"- Stiva sistemului (stack )"<<endl;

cout<<"- Segmental de date (data segment )"<<endl;

cout<<"- Zona de adrese libere ( heap )"<<endl;

cout<<" Variabilele de memorie pot fi clasificate dupa urmatoarele doua criterii:"<<endl;

cout<<"1) Durata de viata reprezinta perioada de timp in care variabilelor li se aloca spatiu in memoria interna."<<endl;

cout<<"2) Domeniul de vizibilitate reprezinta zona din program in care este permis"<<endl;

cout<<"accesul la variabile."<<endl;

cout<<"1) Dupa duarata de viata:"<<endl;

cout<<"a. Variabile cu durata locala:"<<endl;

cout<<"- Sunt cele declarate in interiorul unui bloc ( subprogram )"<<endl;

cout<<"- Li se rezerva spatiu in stiva, primind la declarare valoarea reziduala"<<endl;

cout<<"( valoarea care se gaseste in stiva in locul in care a fost declarata variabila)"<<endl;

cout<<"- Durata lor de viata coincide cu perioada in care se executa instructiunile din";

cout<<"corpul subprogramului."<<endl;

cout<<"- Nu pot fi accesate inafara blocului in care au fost declarate."<<endl;

cout<<"b. Variabile cu durata static:"<<endl;

cout<<"- Sunt declarate inafara oricarui bloc"<<endl;

cout<<"- Pot fi accesate in toate subprogramele definite fupa declararea lor."<<endl;

cout<<"- Li se rezerva spatiu in segmental de date si sunt initializate in momentul"<<endl;

cout<<"declararii cu valoarea 0."<<endl;

cout<<"c. Variabile cu durata dinamica:"<<endl;

cout<<"- Li se rezerva spatiu doar in momentul in care sunt utilizati."<<endl;

cout<<"- Zona de memorie folosita de aceste variabile este zona de adrese libere."<<endl;

cout<<"- Imediat dupa utilizarea acestora zona de memorie a acestora este eliberata."<<endl;

cout<<"- Sunt variabile care contin adrese si se numesc pointeri."<<endl;

cout<<"- Utilizarea variabilelor dinamice reprezinta cea mai eficienta metode de"<<endl;

cout<<"alocare a memoriei."<<endl;

cout<<"2) Dupa domeniul de vizibilitate:"<<endl;

cout<<"a. Variabile locale( corespund cu variabilele cu durata locala de la primul"<<endl;

cout<<"criteriu )."<<endl;

cout<<"b. Variabile globale (corespund variabilelor cu durata static )."<<endl;

cout<<" Observatie:"<<endl;

cout<<" In cazul in care intr-un program exista o variabila locala si o variabila"<<endl;

cout<<"globala cu aceeasi nume, in subprogramul in care a fost declarata cea locala"<<endl;

cout<<"are intaietate."<<endl;

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5,6 sau 7 pentru intoarcerea la meniul principal!"<<endl;

}

cout<<endl;

cout<<"Meniul pentru teoria de la subprograme: "<<endl;

cout<<"1.Notiuni introductive"<<endl;

cout<<"2.Clasificarea subprogramelor"<<endl;

cout<<"3.Reguli pentru construirea subprogramelor C++"<<endl;

cout<<"4.Utilizarea stivei de catre subprograme"<<endl;

cout<<"5.Tablourile de memorie si subprogramele"<<endl;

cout<<"6.Clasificarea variabilelor de memorie"<<endl;

cout<<"7.Pentru intoarcerea la meniul principal"<<endl;

cout<<"Introduceti cazul dorit: ";

cin>>lectii;

}

}break;

case 2:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor:"<<endl;

cout<<"1.Pentru programele implementate cu ajutorul subprogramelor primite la tema;"<<endl;

cout<<"2.Pentru alte programe implementate cu ajutorul subprogramelor;"<<endl;

cout<<"3.Pentru intoarcearea la meniul principal."<<endl;

cin>>prog;

while(prog!=3)

{

switch(prog)

{

case 1:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema:"<<endl;

cout<<"1.Tema 1;"<<endl;

cout<<"2.Tema 2;"<<endl;

cout<<"3.Pentru intoarcearea la meniul secundar."<<endl;

cin>>prog1;

while(prog1!=3)

{

switch(prog1)

{

case 1:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema1:"<<endl;

cout<<"1.Functii procedurale;"<<endl;

cout<<"2.Functii operand;"<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog2;

while(prog2!=3)

{

switch(prog2)

{

case 1:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema1, functii procedurale:"<<endl;

cout<<"1.Sa se realizeze programe in C++ care, utilizand functii procedurale, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)a+b"<<endl;

cout<<"b)a-b"<<endl;

cout<<"c)a\*b"<<endl;

cout<<"d)a/b"<<endl;

cout<<"Observatie: Se vor utiliza subprograme att pentru calcule, cat si pentru citire si afisare."<<endl;

cout<<endl;

cout<<"2.Sa se realizeze programe in C++ care, utilizand functii procedurale, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)(2(a+b)-(ab-bc))/(pow(a,2)-pow(b,2))"<<endl;

cout<<"b)(2(a+1)-(3b-4))/((a-b)(b+14))"<<endl;

cout<<"Observatie: Se vor utiliza subprograme atat pentru calcule, cat si pentru citire si afisare."<<endl;

cout<<endl;

cout<<"3.Sa se realizeze programe în C++ care, utilizand functii procedurale, permit efectuarea urmatoarelor calcule cu numere complexe:"<<endl;

cout<<"a)z1+z2"<<endl;

cout<<"b)z1-z2"<<endl;

cout<<"c)z1\*z2"<<endl;

cout<<"d)z1/z2"<<endl;

cout<<endl;

cout<<"4.Sa se realizeze programe în C++ care, utilizând functii procedurale, permit efectuarea următoarelor calcule cu fractii:"<<endl;

cout<<"a)f1+f2"<<endl;

cout<<"b)f1-f2"<<endl;

cout<<"c)f1\*f2"<<endl;

cout<<"d)f1/f2"<<endl;

cout<<endl;

cout<<"5.Pentru intoarcearea la meniul anterior"<<endl;

cin>>prog3;

while(prog3!=5)

{

switch(prog3)

{

case 1:

{

int a,b,s,d,p;

float c;

citire(a);

citire(b);

afisare(a);

afisare(b);

suma(a,b,s);

afisare(s);

diferenta(a,b,d);

afisare(d);

produs(a,b,p);

afisare(p);

cat(a,b,c);

afisare(c);

}break;

case 2:

{

cout<<"1.a"<<endl;

cout<<"2.b"<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior"<<endl;

cin>>ab;

while(ab!=3)

{

switch(ab)

{

case 1:

{

unsigned a,b,c,s,p,p1,p3,pa,pb;

int d,d1,pab;

float r;

citire1(a);

citire1(b);

citire1(c);

afisare1(a);

afisare1(b);

afisare1(c);

suma1(a,b,s);

produs1(a,b,p);

produs1(b,c,p1);

diferenta1(p,p1,d);

produs1(2,s,p3);

diferenta1(p3,d,d1);

produs1(a,a,pa);

produs1(b,b,pb);

diferenta1(pa,pb,pab);

cat1(d1,pab,r);

afisare1(r);

}break;

case 2:

{

unsigned a,b,s,s2,p,p1,p3,pj;

int d,d1,ds;

float r;

citire1(a);

citire1(b);

afisare1(a);

afisare1(b);

suma1(a,1,s);

produs1(3,b,p);

diferenta1(p,4,d);

produs1(2,s,p3);

diferenta1(a,b,d1);

suma1(b,14,s2);

produs1(d1,s2,pj);

diferenta1(p3,d,ds);

cat1(ds,pj,r);

afisare1(r);

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"1.a"<<endl;

cout<<"2.b"<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior"<<endl;

cin>>ab;

}

}break;

case 3:

{

citire2(z1);

citire2(z2);

cout<<"Cele 2 numere complexe sunt:"<<endl;

afisare2(z1);

afisare2(z2);

cout<<"Produsul este:"<<endl;

produs2(z1,z2,zp);

afisare2(zp);

cout<<"Raportul este:"<<endl;

raport2(z1,z2,zr);

afisare2(zr);

cout<<"Suma este:"<<endl;

suma2(z1,z2,zs);

afisare2(zs);

cout<<"Diferenta este:"<<endl;

diferenta2(z1,z2,zd);

afisare2(zd);

}break;

case 4:

{

citire3(f1);

citire3(f2);

cout<<"Cele 2 fractii sunt:"<<endl;

afisare3(f1);

afisare3(f2);

cout<<"Produsul este:"<<endl;

produs3(f1,f2,fp);

afisare3(fp);

cout<<"Raportul este:"<<endl;

raport3(f1,f2,fr);

afisare3(fr);

cout<<"Suma este:"<<endl;

suma3(f1,f2,fs);

afisare3(fs);

cout<<"Diferenta este:"<<endl;

diferenta3(f1,f2,fd);

afisare3(fd);

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4 sau 5 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema1, funcţii procedurale:"<<endl;

cout<<"1.Sa se realizeze programe in C++ care, utilizand functii procedurale, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)a+b"<<endl;

cout<<"b)a-b"<<endl;

cout<<"c)a\*b"<<endl;

cout<<"d)a/b"<<endl;

cout<<"Observatie: Se vor utiliza subprograme att pentru calcule, cat si pentru citire si afisare."<<endl;

cout<<endl;

cout<<"2.Sa se realizeze programe in C++ care, utilizand functii procedurale, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)(2(a+b)-(ab-bc))/(pow(a,2)-pow(b,2))"<<endl;

cout<<"b)(2(a+1)-(3b-4))/((a-b)(b+14))"<<endl;

cout<<"Observatie: Se vor utiliza subprograme atat pentru calcule, cat si pentru citire si afisare."<<endl;

cout<<endl;

cout<<"3.Sa se realizeze programe în C++ care, utilizand functii procedurale, permit efectuarea urmatoarelor calcule cu numere complexe:"<<endl;

cout<<"a)z1+z2"<<endl;

cout<<"b)z1-z2"<<endl;

cout<<"c)z1\*z2"<<endl;

cout<<"d)z1/z2"<<endl;

cout<<endl;

cout<<"4.Sa se realizeze programe în C++ care, utilizând functii procedurale, permit efectuarea următoarelor calcule cu fractii:"<<endl;

cout<<"a)f1+f2"<<endl;

cout<<"b)f1-f2"<<endl;

cout<<"c)f1\*f2"<<endl;

cout<<"d)f1/f2"<<endl;

cout<<endl;

cout<<"5.Pentru intoarcearea la meniul anterior"<<endl;

cin>>prog3;

}

}break;

case 2:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema1, functii operand:"<<endl;

cout<<"1.Sa se realizeze programe in C++ care, utilizand functii operand, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)a+b"<<endl;

cout<<"b)a-b"<<endl;

cout<<"c)a\*b"<<endl;

cout<<"d)a/b"<<endl;

cout<<"e)a DIV b"<<endl;

cout<<"f)a MOD b"<<endl;

cout<<"g)(a+b)(cd-2a)+3c"<<endl;

cout<<endl;

cout<<"2.Sa se realizeze programe in C++ care, utilizand functii operand, daca:"<<endl;

cout<<" \* x+1,x<=4 "<<endl;

cout<<"f(x)\* "<<endl;

cout<<" \* 2x-1,x>4 "<<endl;

cout<<" permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)f(a)"<<endl;

cout<<"b)f(a)+f(b)"<<endl;

cout<<endl;

cout<<"3.Sa se realizeze programe in C++ care, utilizand functii operand, daca:"<<endl;

cout<<" \* x+y,x<=y "<<endl;

cout<<"E(x,y)\* "<<endl;

cout<<" \* 2xy-1,x>y "<<endl;

cout<<" permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)E(a,b)"<<endl;

cout<<"b)E(a,b)+E(b,a)"<<endl;

cout<<endl;

cout<<"4.Sa se realizeze programe in C++ care, utilizand functii operand, permit calcularea:"<<endl;

cout<<"a)modulului unui numar"<<endl;

cout<<"b)sumei primelor n numere naturale"<<endl;

cout<<"c)factorialului unui numar"<<endl;

cout<<endl;

cout<<"5.Sa se realizeze programe in C++ care, utilizand functii operand, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<" k"<<endl;

cout<<"a)C"<<endl;

cout<<" n"<<endl;

cout<<" k"<<endl;

cout<<"b)A"<<endl;

cout<<" n"<<endl;

cout<<endl;

cout<<"6.Pentru intoarcearea la meniul anterior"<<endl;

cin>>prog3;

while(prog3!=6)

{

switch(prog3)

{

case 1:

{

unsigned a,b,c,d;

citire4(a);

citire4(b);

citire4(c);

citire4(d);

afisare4(a);

afisare4(b);

afisare4(c);

afisare4(d);

afisare4(suma4(a,b));

afisare4(diferenta4(a,b));

afisare4(produs4(a,b));

afisare4(cat4(a,b));

afisare4(div4(a,b));

afisare4(mod4(a,b));

afisare4(calcul4(a,b,c,d));

}break;

case 2:

{

int a,b;

cout<<"a=";

cin>>a;

cout<<"b=";

cin>>b;

cout<<"f("<<a<<")="<<fct(a)<<endl;

cout<<"f("<<a<<")"<<"+f("<<b<<")="<<fct(a)+fct(b);

}break;

case 3:

{

int a,b;

cout<<"a=";

cin>>a;

cout<<"b=";

cin>>b;

cout<<"E("<<a<<","<<b<<")="<<E(a,b)<<endl;

cout<<"E("<<a<<","<<b<<")+E("<<b<<","<<a<<")="<<E(a,b)+E(b,a);

}break;

case 4:

{

int x;

cout<<"x=";

cin>>x;

cout<<"Factorialul="<<factorial5(x)<<endl;

cout<<"Modulul="<<modul5(x)<<endl;

cout<<"Suma primelor "<<x<<" numere naturale este:"<<suma5(x);

}break;

case 5:

{

int m,k;

cout<<"n=";

cin>>m;

cout<<"k=";

cin>>k;

cout<<factorial(m)<<endl;

cout<<aranjamente(m,k)<<endl;

cout<<combinari(m,k)<<endl;

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5 sau 6 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema1, functii operand:"<<endl;

cout<<"1.Sa se realizeze programe in C++ care, utilizand functii operand, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)a+b"<<endl;

cout<<"b)a-b"<<endl;

cout<<"c)a\*b"<<endl;

cout<<"d)a/b"<<endl;

cout<<"e)a DIV b"<<endl;

cout<<"f)a MOD b"<<endl;

cout<<"g)(a+b)(cd-2a)+3c"<<endl;

cout<<endl;

cout<<"2.Sa se realizeze programe in C++ care, utilizand functii operand, daca:"<<endl;

cout<<" \* x+1,x<=4 "<<endl;

cout<<"f(x)\* "<<endl;

cout<<" \* 2x-1,x>4 "<<endl;

cout<<" permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)f(a)"<<endl;

cout<<"b)f(a)+f(b)"<<endl;

cout<<endl;

cout<<"3.Sa se realizeze programe in C++ care, utilizand functii operand, daca:"<<endl;

cout<<" \* x+y,x<=y "<<endl;

cout<<"E(x,y)\* "<<endl;

cout<<" \* 2xy-1,x>y "<<endl;

cout<<" permit efectuarea urmatoarelor calcule:"<<endl;

cout<<"a)E(a,b)"<<endl;

cout<<"b)E(a,b)+E(b,a)"<<endl;

cout<<endl;

cout<<"4.Sa se realizeze programe in C++ care, utilizand functii operand, permit calcularea:"<<endl;

cout<<"a)modulului unui numar"<<endl;

cout<<"b)sumei primelor n numere naturale"<<endl;

cout<<"c)factorialului unui numar"<<endl;

cout<<endl;

cout<<"5.Sa se realizeze programe in C++ care, utilizand functii operand, permit efectuarea urmatoarelor calcule:"<<endl;

cout<<" k"<<endl;

cout<<"a)C"<<endl;

cout<<" n"<<endl;

cout<<" k"<<endl;

cout<<"b)A"<<endl;

cout<<" n"<<endl;

cout<<endl;

cout<<"6.Pentru intoarcearea la meniul anterior"<<endl;

cin>>prog3;

}

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema1:"<<endl;

cout<<"1.Functii procedurale;"<<endl;

cout<<"2.Functii operand;"<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog2;

}

}break;

case 2:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema2:"<<endl;

cout<<"1.Tablouri de memorie si subprograme;"<<endl;

cout<<"2.Programe diverse;"<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog2;

while(prog2!=3)

{

switch(prog2)

{

case 1:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema2, tablouri de memorie si subprograme:"<<endl;

cout<<"1.Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu vectori:"<<endl;

cout<<"a)citirea si afisarea unui vector;"<<endl;

cout<<"b)citirea si afisarea a doi vectori si calcularea sumei acestora;"<<endl;

cout<<"c)citirea si afisarea unui vector si calcularea sumei elementelor acestuia;"<<endl;

cout<<"d)citirea unui vector si afişarea elementelor acestuia in ordine inversa;"<<endl;

cout<<"e)citirea si afisarea a doi vectori si calcularea produsului scalar al acestora;"<<endl;

cout<<"f)citirea si afisarea a doi vectori si calcularea produsului vectorial al acestora;"<<endl;

cout<<"g)citirea si afisarea unui vector si determinarea maximului dintre elementele sale;"<<endl;

cout<<"h)citirea si afisarea unui vector si determinarea minimului dintre elementele sale;"<<endl;

cout<<"i)citirea si afisarea unui vector si sortarea elementelor sale crescator utilizand metoda selectiei directe;"<<endl;

cout<<"j)citirea si afisarea unui vector si sortarea elementelor sale descrescator utilizand metoda bulelor."<<endl;

cout<<endl;

cout<<"2.Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu matrice patrate:"<<endl;

cout<<"a)citirea si afisarea unei matrice;"<<endl;

cout<<"b)citirea si afisarea a două matrice si calcularea sumei acestora;"<<endl;

cout<<"c)citirea si afisarea a două matrice si calcularea produsului acestora;"<<endl;

cout<<"d)citirea si afisarea unei matrice si calcularea produsului dintre aceasta si un scalar;"<<endl;

cout<<"e)citirea si afisarea unei matrice si calcularea sumei elementelor sale;"<<endl;

cout<<"f)citirea si afisarea unei matrice si calcularea sumei elementelor de pe diagonala principala;"<<endl;

cout<<"g)citirea si afisarea unei matrice si calcularea produsului elementelor de deasupra diagonalei principale (p1) si produsului elementelor de sub diagonala principala (p2);"<<endl;

cout<<"h)citirea si afisarea unei matrice si determinarea maximului dintre elementele sale;"<<endl;

cout<<"i)citirea si afisarea unei matrice si determinarea minimului dintre elementele sale."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

while(prog3!=3)

{

switch(prog3)

{

case 1:

{

cout<<"Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu vectori:"<<endl;

cout<<"1.citirea si afisarea unui vector;"<<endl;

cout<<"2.citirea si afisarea a doi vectori si calcularea sumei acestora;"<<endl;

cout<<"3.citirea si afisarea unui vector si calcularea sumei elementelor acestuia;"<<endl;

cout<<"4.citirea unui vector si afişarea elementelor acestuia in ordine inversa;"<<endl;

cout<<"5.citirea si afisarea a doi vectori si calcularea produsului scalar al acestora;"<<endl;

cout<<"6.citirea si afisarea a doi vectori si calcularea produsului vectorial al acestora;"<<endl;

cout<<"7.citirea si afisarea unui vector si determinarea maximului dintre elementele sale;"<<endl;

cout<<"8.citirea si afisarea unui vector si determinarea minimului dintre elementele sale;"<<endl;

cout<<"9.citirea si afisarea unui vector si sortarea elementelor sale crescator utilizand metoda selectiei directe;"<<endl;

cout<<"10.citirea si afisarea unui vector si sortarea elementelor sale descrescator utilizand metoda bulelor."<<endl;

cout<<"11.Pentru intoarcearea la meniul anterior"<<endl;

cin>>ab;

while(ab!=11)

{

switch(ab)

{

case 1:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire6(n,x);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare6(n,x);

}break;

case 2:

{

int x[100], y[100], s[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire7(n,x);

cout<<"Introduceti elementele vectorului y:"<<endl;

citire7(n,y);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare7(n,x);

cout<<"Elementele vectorului y sunt:"<<endl;

afisare7(n,y);

cout<<"Suma acestora este:"<<endl;

suma7(n,x,y,s);

afisare7(n,s);

}break;

case 3:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire8(n,x);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare8(n,x);

cout<<"Suma elementelor vectorului este:"<<endl;

cout<<suma8(n,x);

}break;

case 4:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire9(n,x);

cout<<"Elementele vectorului x in ordine inversa sunt:"<<endl;

afisare9(n,x);

}break;

case 5:

{

int x[100], y[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire10(n,x);

cout<<"Introduceti elementele vectorului y:"<<endl;

citire10(n,y);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare10(n,x);

cout<<"Elementele vectorului y sunt:"<<endl;

afisare10(n,y);

cout<<"Produsul scalar al acestora este:"<<endl;

cout<<produs10(n,x,y);

}break;

case 6:

{

int x[100], y[100], pv[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire11(n,x);

cout<<"Introduceti elementele vectorului y:"<<endl;

citire11(n,y);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare11(n,x);

cout<<"Elementele vectorului y sunt:"<<endl;

afisare11(n,y);

cout<<"Produsul vectorial al acestora este:"<<endl;

produs11(n,x,y,pv);

afisare11(n,pv);

}break;

case 7:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire12(n,x);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare12(n,x);

cout<<"Maximul dintre elementele vectorului x este:"<<endl;

cout<<max12(n,x);

}break;

case 8:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire13(n,x);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare13(n,x);

cout<<"Minimul dintre elementele vectorului x este:"<<endl;

cout<<min13(n,x);

}break;

case 9:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire15(n,x);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare15(n,x);

cout<<"Elementele sortate ale vectorului x sunt:"<<endl;

sortare15(n,x);

afisare15(n,x);

}break;

case 10:

{

int x[100];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele vectorului x:"<<endl;

citire16(n,x);

cout<<"Elementele vectorului x sunt:"<<endl;

afisare16(n,x);

cout<<"Elementele sortate ale vectorului x sunt:"<<endl;

bule16(n,x);

afisare16(n,x);

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5,6,7,8,9,10 sau 11 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu vectori:"<<endl;

cout<<"1.citirea si afisarea unui vector;"<<endl;

cout<<"2.citirea si afisarea a doi vectori si calcularea sumei acestora;"<<endl;

cout<<"3.citirea si afisarea unui vector si calcularea sumei elementelor acestuia;"<<endl;

cout<<"4.citirea unui vector si afişarea elementelor acestuia in ordine inversa;"<<endl;

cout<<"5.citirea si afisarea a doi vectori si calcularea produsului scalar al acestora;"<<endl;

cout<<"6.citirea si afisarea a doi vectori si calcularea produsului vectorial al acestora;"<<endl;

cout<<"7.citirea si afisarea unui vector si determinarea maximului dintre elementele sale;"<<endl;

cout<<"8.citirea si afisarea unui vector si determinarea minimului dintre elementele sale;"<<endl;

cout<<"9.citirea si afisarea unui vector si sortarea elementelor sale crescator utilizand metoda selectiei directe;"<<endl;

cout<<"10.citirea si afisarea unui vector si sortarea elementelor sale descrescator utilizand metoda bulelor."<<endl;

cout<<"11.Pentru intoarcearea la meniul anterior"<<endl;

cin>>ab;

}

}break;

case 2:

{

cout<<"Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu matrice patrate:"<<endl;

cout<<"1.citirea si afisarea unei matrice;"<<endl;

cout<<"2.citirea si afisarea a două matrice si calcularea sumei acestora;"<<endl;

cout<<"3.citirea si afisarea a două matrice si calcularea produsului acestora;"<<endl;

cout<<"4.citirea si afisarea unei matrice si calcularea produsului dintre aceasta si un scalar;"<<endl;

cout<<"5.citirea si afisarea unei matrice si calcularea sumei elementelor sale;"<<endl;

cout<<"6.citirea si afisarea unei matrice si calcularea sumei elementelor de pe diagonala principala;"<<endl;

cout<<"7.citirea si afisarea unei matrice si calcularea produsului elementelor de deasupra diagonalei principale (p1) si produsului elementelor de sub diagonala principala (p2);"<<endl;

cout<<"8.citirea si afisarea unei matrice si determinarea maximului dintre elementele sale;"<<endl;

cout<<"9.citirea si afisarea unei matrice si determinarea minimului dintre elementele sale."<<endl;

cout<<"10.Pentru intoarcearea la meniul anterior"<<endl;

cin>>ab;

while(ab!=10)

{

switch(ab)

{

case 1:

{

int x[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire17(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare17(n,x);

return 0;

}break;

case 2:

{

int x[20][20],y[20][20],s[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire117(n,x);

cout<<"Introduceti elementele matricei y:"<<endl;

citire117(n,y);

cout<<"Elementele matricei x sunt:"<<endl;

afisare117(n,x);

cout<<"Elementele matricei y sunt:"<<endl;

afisare117(n,y);

cout<<"Suma acestora este:"<<endl;

suma117(n,x,y,s);

afisare117(n,s);

}break;

case 3:

{

int x[20][20],y[20][20],p[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire18(n,x);

cout<<"Introduceti elementele matricei y:"<<endl;

citire18(n,y);

cout<<"Elementele matricei x sunt:"<<endl;

afisare18(n,x);

cout<<"Elementele matricei y sunt:"<<endl;

afisare18(n,y);

cout<<"Produsul acestora este:"<<endl;

produs18(n,x,y,p);

afisare18(n,p);

}break;

case 4:

{

int x[20][20],y[20][20],p[20][20];

unsigned n,sc;

cout<<"n=";

cin>>n;

cout<<"sc=";

cin>>sc;

cout<<"Introduceti elementele matricei x:"<<endl;

citire19(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare19(n,x);

cout<<"Produsul dintre matrice si un scalar este:"<<endl;

produs19(n,sc,x,p);

afisare19(n,p);

}break;

case 5:

{

int x[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire20(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare20(n,x);

cout<<"Suma elementelor este:"<<endl;

cout<<suma20(n,x);

}break;

case 6:

{

int x[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire21(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare21(n,x);

cout<<"Suma elementelor de pe diagonala principala este:"<<endl;

cout<<suma21(n,x);

}break;

case 7:

{

int x[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire111(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare111(n,x);

cout<<"Produsului elementelor de deasupra diagonalei principale este:"<<endl;

cout<<"p1="<<produs111(n,x)<<endl;

cout<<"Produsului elementelor de sub diagonala principala este:"<<endl;

cout<<"p2="<<produs222(n,x);

}break;

case 8:

{

int x[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire22(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare22(n,x);

cout<<"Maximul dintre elementelor este:"<<endl;

cout<<maximul22(n,x);

}break;

case 9:

{

int x[20][20];

unsigned n;

cout<<"n=";

cin>>n;

cout<<"Introduceti elementele matricei x:"<<endl;

citire23(n,x);

cout<<"Elementele matricei x sunt:"<<endl;

afisare23(n,x);

cout<<"Minimul dintre elementelor este:"<<endl;

cout<<minimul23(n,x);

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5,6,7,8,9 sau 10 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu matrice patrate:"<<endl;

cout<<"1.citirea si afisarea unei matrice;"<<endl;

cout<<"2.citirea si afisarea a două matrice si calcularea sumei acestora;"<<endl;

cout<<"3.citirea si afisarea a două matrice si calcularea produsului acestora;"<<endl;

cout<<"4.citirea si afisarea unei matrice si calcularea produsului dintre aceasta si un scalar;"<<endl;

cout<<"5.citirea si afisarea unei matrice si calcularea sumei elementelor sale;"<<endl;

cout<<"6.citirea si afisarea unei matrice si calcularea sumei elementelor de pe diagonala principala;"<<endl;

cout<<"7.citirea si afisarea unei matrice si calcularea produsului elementelor de deasupra diagonalei principale (p1) si produsului elementelor de sub diagonala principala (p2);"<<endl;

cout<<"8.citirea si afisarea unei matrice si determinarea maximului dintre elementele sale;"<<endl;

cout<<"9.citirea si afisarea unei matrice si determinarea minimului dintre elementele sale."<<endl;

cout<<"10.Pentru intoarcearea la meniul anterior"<<endl;

cin>>ab;

}

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema2, tablouri de memorie si subprograme:"<<endl;

cout<<"1.Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu vectori:"<<endl;

cout<<"a)citirea si afisarea unui vector;"<<endl;

cout<<"b)citirea si afisarea a doi vectori si calcularea sumei acestora;"<<endl;

cout<<"c)citirea si afisarea unui vector si calcularea sumei elementelor acestuia;"<<endl;

cout<<"d)citirea unui vector si afişarea elementelor acestuia in ordine inversa;"<<endl;

cout<<"e)citirea si afisarea a doi vectori si calcularea produsului scalar al acestora;"<<endl;

cout<<"f)citirea si afisarea a doi vectori si calcularea produsului vectorial al acestora;"<<endl;

cout<<"g)citirea si afisarea unui vector si determinarea maximului dintre elementele sale;"<<endl;

cout<<"h)citirea si afisarea unui vector si determinarea minimului dintre elementele sale;"<<endl;

cout<<"i)citirea si afisarea unui vector si sortarea elementelor sale crescator utilizand metoda selectiei directe;"<<endl;

cout<<"j)citirea si afisarea unui vector si sortarea elementelor sale descrescator utilizand metoda bulelor."<<endl;

cout<<endl;

cout<<"2.Sa se realizeze programe in C++ care, utilizand functii, permit efectuarea urmatoarelor calcule cu matrice patrate:"<<endl;

cout<<"a)citirea si afisarea unei matrice;"<<endl;

cout<<"b)citirea si afisarea a două matrice si calcularea sumei acestora;"<<endl;

cout<<"c)citirea si afisarea a două matrice si calcularea produsului acestora;"<<endl;

cout<<"d)citirea si afisarea unei matrice si calcularea produsului dintre aceasta si un scalar;"<<endl;

cout<<"e)citirea si afisarea unei matrice si calcularea sumei elementelor sale;"<<endl;

cout<<"f)citirea si afisarea unei matrice si calcularea sumei elementelor de pe diagonala principala;"<<endl;

cout<<"g)citirea si afisarea unei matrice si calcularea produsului elementelor de deasupra diagonalei principale (p1) si produsului elementelor de sub diagonala principala (p2);"<<endl;

cout<<"h)citirea si afisarea unei matrice si determinarea maximului dintre elementele sale;"<<endl;

cout<<"i)citirea si afisarea unei matrice si determinarea minimului dintre elementele sale."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

}

}break;

case 2:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema2, programe diverse:"<<endl;

cout<<"1.Program pentru rezolvarea ecuatiei de gradul al doilea utilizand subprograme."<<endl;

cout<<"Varianta I: utilizand parametri"<<endl;

cout<<"Varianta a II-a: utilizand variabile globale"<<endl;

cout<<endl;

cout<<"2.Realizati un program C++ care permite citirea unui vector cu elemente numere reale, reprezentand temperaturile corespunzatoare celor douasprezece luni ale unui an. In functie de dorinta utilizatorului si ori de cate ori doreste acesta sa se determine:"<<endl;

cout<<"1.Temperatura maxima"<<endl;

cout<<"2.Temperatura minima"<<endl;

cout<<"3.Temperatura medie"<<endl;

cout<<"4.Ordonarea crescatoare a temperaturilor utilizand metoda selectiei directe"<<endl;

cout<<"5.Ordonarea descrescatoare a temperaturilor utilizand metoda bulelor"<<endl;

cout<<"6.Oprirea programului"<<endl;

cout<<"Citirea unei alte optiuni în afara de 1, 2, 3, 4, 5 sau 6 determina afisarea mesajului “Optiune gresita!”"<<endl;

cout<<endl;

cout<<"3.In fisierul vector.txt se afla pe primul rand un numar natural n, iar pe al doilea rand n numerele naturale despartite printr-un spatiu. Se considera ca n reprezinta dimensiunea unui vector, iar cele n elemente, elementele vectorului. Sa se determine suma elementele din vector care sunt palindrom si sa se afiseze pe ecran si in fisierul suma.txt. Pentru citirea vectorului si verificarea proprietatii de palindrom se vor utiliza subprogram."<<endl;

cout<<endl;

cout<<"4.Afisati pe ecran cate elemente din vector au exact p divizori proprii, unde p este un numar natural citit de la tastatura."<<endl;

cout<<endl;

cout<<"5.Afisati cate numere perfecte sunt in vector, afisarea se face in fisierul solutie.out si pe ecran.";

cout<<endl;

cout<<"6.Afisati vectorul si valoarea elementului minim din vector precum si pozitiile pe care acesta apare in vector. Pentru citirea si afisarea vectorului si determinantului minimul se vor utiliza subprograme."<<endl;

cout<<"7.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

while(prog3!=7)

{

switch(prog3)

{

case 1:

{

cout<<"Program pentru rezolvarea ecuatiei de gradul al doilea utilizand subprograme:"<<endl;

cout<<"1.Varianta I: utilizand parametri."<<endl;

cout<<"2.Varianta a II-a: utilizand variabile globale."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>ab;

while(ab!=11)

{

switch(ab)

{

case 1:

{

float a,b,c;

cout<<"a=";

cin>>a;

cout<<"b=";

cin>>b;

cout<<"c=";

cin>>c;

if(a==0)

if(b==0)

if(c==0)

cout<<"x poate lua orice valoare.";

else

cout<<"Caz imposibil.";

else

{

cout<<"Ecuatia este de gradul I.";

cout<<"x="<<ecuatie1199(b,c);

}

else

{

cout<<"Ecuatia este de gradul II."<<endl;

cout<<"Delta="<<delta199(a,b,c)<<endl;

if(delta199(a,b,c)>0)

caz1199(a,b,c);

else if(delta199(a,b,c)==0)

caz2199(a,b,c);

else

caz3199(a,b,c);

}

}break;

case 2:

{

cout<<"a=";

cin>>a;

cout<<"b=";

cin>>b;

cout<<"c=";

cin>>c;

if(a==0)

if(b==0)

if(c==0)

cout<<"x poate lua orice valoare.";

else

cout<<"Caz imposibil.";

else

{

cout<<"Ecuatia este de gradul I.";

cout<<"x="<<ecuatie1198();

}

else

{

cout<<"Ecuatia este de gradul II."<<endl;

cout<<"Delta="<<delta198()<<endl;

if(delta198()>0)

caz1198();

else if(delta198()==0)

caz2198();

else

caz3198();

}

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Program pentru rezolvarea ecuatiei de gradul al doilea utilizand subprograme:"<<endl;

cout<<"1.Varianta I: utilizand parametri."<<endl;

cout<<"2.Varianta a II-a: utilizand variabile globale."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>ab;

}

}break;

case 2:

{

int x[12],n,i;

cout<<"Introduceti temperaturile pe cele 12 luni:"<<endl;

citire100(x);

cout<<"Alegeti dintre urmatoarele optiuni:"<<endl;

cout<<"1.Temperatura maxima;"<<endl;

cout<<"2.Temperatura minima;"<<endl;

cout<<"3.Temperatura medie;"<<endl;

cout<<"4.Ordonarea crescatoare a temperaturilor utilizand metoda selectiei directe;"<<endl;

cout<<"5.Ordonarea descrescatoare a temperaturilor utilizand metoda bulelor;"<<endl;

cout<<"6.Oprirea programului."<<endl;

cin>>n;

while(n!=6)

{

switch(n)

{

case 1:

{

cout<<max100(x)<<endl;

}

break;

case 2:

{

cout<<min100(x)<<endl;

}

break;

case 3:

{

cout<<media100(x)<<endl;

}

break;

case 4:

{

sortare100(x);

for(i=1;i<=12;i++)

cout<<x[i]<<" ";

cout<<endl;

}

break;

case 5:

{

bule100(x);

for(i=1;i<=12;i++)

cout<<x[i]<<" ";

cout<<endl;

}

break;

default: cout<<"Optiune gresita!"<<endl;

}

cout<<"Alegeti o alta optiune:";

cin>>n;

}

}break;

case 3:

{

unsigned a[100],n,i,s=0;

f11>>n;

citire24(n,a);

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

{

if(palindrom24(a[i])==1)

s=s+a[i];

}

if(s==0)

cout<<"Nu exista numere palindrom."<<endl;

else

{

cout<<s;

f22<<s;

}

}break;

case 4:

{

unsigned a[100],n,i,p,s=0;

f111>>n;

citire25(n,a);

cout<<"p=";

cin>>p;

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

if(divizor25(a[i])==p)

s++;

cout<<s;

}break;

case 5:

{

unsigned a[100],n,i,p,s=0;

f1111>>n;

citire26(n,a);

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

if(perfect26(a[i])==1)

s=s+1;

cout<<s;

f2222<<s;

}break;

case 6:

{

int a[100],n,i;

f11111>>n;

citire27(n,a);

afisare27(n,a);

cout<<"Minimul este "<<min27(n,a)<<" si se afla pe pozitia:";

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

if(a[i]==min27(n,a))

cout<<i<<" ";

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5,6 sau 7 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema2, programe diverse:"<<endl;

cout<<"1.Program pentru rezolvarea ecuatiei de gradul al doilea utilizand subprograme."<<endl;

cout<<"Varianta I: utilizand parametri"<<endl;

cout<<"Varianta a II-a: utilizand variabile globale"<<endl;

cout<<endl;

cout<<"2.Realizati un program C++ care permite citirea unui vector cu elemente numere reale, reprezentand temperaturile corespunzatoare celor douasprezece luni ale unui an. In functie de dorinta utilizatorului si ori de cate ori doreste acesta sa se determine:"<<endl;

cout<<"1.Temperatura maxima"<<endl;

cout<<"2.Temperatura minima"<<endl;

cout<<"3.Temperatura medie"<<endl;

cout<<"4.Ordonarea crescatoare a temperaturilor utilizand metoda selectiei directe"<<endl;

cout<<"5.Ordonarea descrescatoare a temperaturilor utilizand metoda bulelor"<<endl;

cout<<"6.Oprirea programului"<<endl;

cout<<"Citirea unei alte optiuni în afara de 1, 2, 3, 4, 5 sau 6 determina afisarea mesajului “Optiune gresita!”"<<endl;

cout<<endl;

cout<<"3.In fisierul vector.txt se afla pe primul rand un numar natural n, iar pe al doilea rand n numerele naturale despartite printr-un spatiu. Se considera ca n reprezinta dimensiunea unui vector, iar cele n elemente, elementele vectorului. Sa se determine suma elementele din vector care sunt palindrom si sa se afiseze pe ecran si in fisierul suma.txt. Pentru citirea vectorului si verificarea proprietatii de palindrom se vor utiliza subprogram."<<endl;

cout<<endl;

cout<<"4.Afisati pe ecran cate elemente din vector au exact p divizori proprii, unde p este un numar natural citit de la tastatura."<<endl;

cout<<endl;

cout<<"5.Afisati cate numere perfecte sunt in vector, afisarea se face in fisierul solutie.out si pe ecran.";

cout<<endl;

cout<<"6.Afisati vectorul si valoarea elementului minim din vector precum si pozitiile pe care acesta apare in vector. Pentru citirea si afisarea vectorului si determinantului minimul se vor utiliza subprograme."<<endl;

cout<<"7.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

}

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema2:"<<endl;

cout<<"1.Tablouri de memorie si subprograme;"<<endl;

cout<<"2.Programe diverse;"<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog2;

}

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul secundar."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor primite la tema:"<<endl;

cout<<"1.Tema 1;"<<endl;

cout<<"2.Tema 2;"<<endl;

cout<<"3.Pentru intoarcearea la meniul secundar."<<endl;

cin>>prog1;

}

}break;

case 2:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor:"<<endl;

cout<<"1.Algoritmi elementari;"<<endl;

cout<<"2.Pentru intoarcearea la meniul secundar."<<endl;

cin>>prog1;

while(prog1!=2)

{

switch(prog1)

{

case 1:

{

cout<<"Meniul programelor implementate cu ajutorul subprogramelor, algoritmi elementari:"<<endl;

cout<<"1.Algoritmi pentru interschimbarea valorilor a doua variabile;"<<endl;

cout<<"2.Algoritmi pentru determinarea maximului si minimului;"<<endl;

cout<<"3.Algoritmi pentru prelucrarea cifrelor unui numar;"<<endl;

cout<<"4.Algoritmi pentru determinarea celui mai mare divizor comun a doua numere;"<<endl;

cout<<"5.Algoritmul pentru testarea unui numar prim;"<<endl;

cout<<"6.Algoritmi pentru extragerea (generarea) divizorilor unui numar;"<<endl;

cout<<"7.Algoritmul pentru generarea sirurilor recurente;"<<endl;

cout<<"8.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog2;

while(prog2!=8)

{

switch(prog2)

{

case 1:

{

int x,y;

cin>>x>>y;

inte(x,y);

cout<<x<<" "<<y;

}break;

case 2:

{

cout<<"Algoritmi pentru determinarea maximului si minimului;"<<endl;

cout<<"1.Determinarea maximului."<<endl;

cout<<"2.Determinarea minimului."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

while(prog3!=3)

{

switch(prog3)

{

case 1:

{

unsigned m;

int x;

cout<<"m=";

cin>>m;

cin>>x;

cout<<"Maximul="<<max31(m,x);

}break;

case 2:

{

int x;

cout<<"Introduceti numerele:"<<endl;

cin>>x;

cout<<"Minimul="<<min32(x);

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Algoritmi pentru determinarea maximului si minimului;"<<endl;

cout<<"1.Determinarea maximului."<<endl;

cout<<"2.Determinarea minimului."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

}

}break;

case 3:

{

cout<<"Algoritmi pentru prelucrarea cifrelor unui numar;"<<endl;

cout<<"1.Algoritmul pentru extragerea cifrelor unui numar."<<endl;

cout<<"2.Algoritmul pentru compunerea unui numar din cifrele sale."<<endl;

cout<<"3.Algoritmul pentru determinarea inversului."<<endl;

cout<<"4.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

while(prog3!=3)

{

switch(prog3)

{

case 1:

{

int m,c;

cout<<"m=";

cin>>m;

extragerea33(m,c);

}break;

case 2:

{

int c;

cout<<"c=";

cin>>c;

cout<<"nr="<<compunerea34(c);

}break;

case 3:

{

int m;

cout<<"m=";

cin>>m;

cout<<"Inv="<<inversul35(m);

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3 sau 4 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Algoritmi pentru prelucrarea cifrelor unui numar;"<<endl;

cout<<"1.Algoritmul pentru extragerea cifrelor unui numar."<<endl;

cout<<"2.Algoritmul pentru compunerea unui numar din cifrele sale."<<endl;

cout<<"3.Algoritmul pentru determinarea inversului."<<endl;

cout<<"4.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

}

}break;

case 4:

{

cout<<"Algoritmi pentru determinarea celui mai mare divizor comun a doua numere;"<<endl;

cout<<"1.Varianta I."<<endl;

cout<<"2.Varianta II."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

while(prog3!=3)

{

switch(prog3)

{

case 1:

{

int x,y;

cout<<"x=";

cin>>x;

cout<<"y=";

cin>>y;

cout<<"cmmdc="<<cmmdc36(x,y);

}break;

case 2:

{

int x,y;

cout<<"x=";

cin>>x;

cout<<"y=";

cin>>y;

cout<<"cmmdc="<<cmmdc37(x,y);

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Algoritmi pentru determinarea celui mai mare divizor comun a doua numere;"<<endl;

cout<<"1.Varianta I."<<endl;

cout<<"2.Varianta II."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

}

}break;

case 5:

{

int m1,m2;

cout<<"m1=";

cin>>m1;

cout<<"m2=";

cin>>m2;

if(prim41(m1)==1)

cout<<"Numarul m1 este prim.";

else

cout<<"Numarul m1 nu este prim.";

cout<<endl;

if(prim41(m2)==1)

cout<<"Numarul m2 este prim.";

else

cout<<"Numarul m2 nu este prim.";

}break;

case 6:

{

cout<<"Algoritmi pentru extragerea (generarea) divizorilor unui numar;"<<endl;

cout<<"1.Varianta I."<<endl;

cout<<"2.Varianta II."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

while(prog3!=3)

{

switch(prog3)

{

case 1:

{

unsigned m;

cout<<"m=";

cin>>m;

divizor42(m);

}break;

case 2:

{

unsigned m;

cout<<"m=";

cin>>m;

divizor43(m);

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Algoritmi pentru extragerea (generarea) divizorilor unui numar;"<<endl;

cout<<"1.Varianta I."<<endl;

cout<<"2.Varianta II."<<endl;

cout<<"3.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog3;

}

}break;

case 7:

{

int m;

cout<<"m=";

cin>>m;

recurente44(m);

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2,3,4,5,6,7 sau 8 pentru intoarcearea la meniul anterior."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor, algoritmi elementari:"<<endl;

cout<<"1.Algoritmi pentru interschimbarea valorilor a doua variabile;"<<endl;

cout<<"2.Algoritmi pentru determinarea maximului si minimului;"<<endl;

cout<<"3.Algoritmi pentru prelucrarea cifrelor unui numar;"<<endl;

cout<<"4.Algoritmi pentru determinarea celui mai mare divizor comun a doua numere;"<<endl;

cout<<"5.Algoritmul pentru testarea unui numar prim;"<<endl;

cout<<"6.Algoritmi pentru extragerea (generarea) divizorilor unui numar;"<<endl;

cout<<"7.Algoritmul pentru generarea sirurilor recurente;"<<endl;

cout<<"8.Pentru intoarcearea la meniul anterior."<<endl;

cin>>prog2;

}

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1 sau 2 pentru intoarcearea la meniul secundar."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor:"<<endl;

cout<<"1.Algoritmi elementari;"<<endl;

cout<<"2.Pentru intoarcearea la meniul secundar."<<endl;

cin>>prog1;

}

}break;

default: cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru intoarcearea la meniul principal."<<endl;

}

cout<<"Meniul programelor implementate cu ajutorul subprogramelor:"<<endl;

cout<<"1.Pentru programele implementate cu ajutorul subprogramelor primite la tema;"<<endl;

cout<<"2.Pentru alte programe implementate cu ajutorul subprogramelor;"<<endl;

cout<<"3.Pentru intoarcearea la meniul principal."<<endl;

cin>>prog;

}

}break;

default:cout<<"Ati introdus un numar gresit. Trebuia sa introduceti 1,2 sau 3 pentru terminarea programului."<<endl;

}

cout<<"Introduceti numarul cazului:"<<endl;

cout<<"1.Pentru teoria de la subprograme;"<<endl;

cout<<"2.Pentru programe implementate cu ajutorul subprogramelor;"<<endl;

cout<<"3.Pentru terminarea programului."<<endl;

cin>>cazul;

}

return 0;

}