Cuprins

[1. Despre tema atestatului 3](#_Toc388305486)

[2. Generarea modelelor fractale 5](#_Toc388305487)

[2.1 Despre modele fractale 5](#_Toc388305488)

[2.2 Curba lui Koch 6](#_Toc388305489)

[2.3 Curba lui Peano 9](#_Toc388305490)

[2.4 Curba C a lui Lévy 10](#_Toc388305491)

[2.5 Curba Dragonului 10](#_Toc388305492)

[3. Funcţia de gradul al II-lea 11](#_Toc388305493)

[3.1 Rezolvare 11](#_Toc388305494)

[3.1.1 Originea Cuvantului 11](#_Toc388305495)

[3.1.2 Radacini 12](#_Toc388305496)

[3.1.3 Forme de exprimare a functiilor de gradul al doilea 12](#_Toc388305497)

[3.2 Reprezentare grafica 13](#_Toc388305498)

[4. Jocul trandafirului 15](#_Toc388305499)

[4.1 Scurt istoric 15](#_Toc388305500)

[4.2 Cum se joaca 15](#_Toc388305501)

[5. Bibliografie 16](#_Toc388305502)

[6. Anexa - codul sursă al aplicaţiei 17](#_Toc388305503)

# Despre tema atestatului

Grafica pe calculator este unul din domeniile informatici preferat in mod special de mine, datorita complexitatii dezvoltarii acestuia.

Grafica computerizata, este in ziua de azi, unul dintre cele mai infloritoare domenii ale informatici interfata de utilizator, GUI, este intr-o continua expansiune, jocurile sunt si ele intr-o continua crestere, cat despre filmele bogate in efecte, de la cele mai simple pana la cele 3D, cresterea cererii este incontestabila.

Dar sa definim putin mai in detaliu domeniul si tema atestatului despre care vom vorbi in continuare.

*Grafica digitală* este un domeniu al informaticii care acoperă toate aspectele legate de prelucrarea imaginilor utilizând un computer. Termenul englez corespunzător este *computer graphics*. Grafica digitală mai este numită uneori grafică de computer, grafică de calculator, grafică pe calculator, grafică computerizată.

Grafica digitală este o activitate în care computerul este utilizat pentru sintetizarea, modificarea, stocarea și managementul imaginilor, precum și pentru prelucrarea informației vizuale obținute din realitatea ce ne înconjoară.

La începutul istoriei sale prin anii ´190 grafica digitală făcea primii pași de progrese tehnice, grafica la calculator fiind destul de limitată în posibilități tehnice. Inițial a dat posibilitatea de a afișa pe ecran doar câteva zeci de segmente și elemente geometrice; se putea lucra doar cu elemente grafice relativ simple.

Evoluțiile, elaborările softului și posibilităților tehnice erau analizate în academii și mediul științific. Treptat însă grafica de calculator s-a stabilit în viața cotidiană, oferind posibilități de realizare a proiectelor comerciale de succes.

Etape în dezvoltarea graficii digitale:

* În anii 1960-1970 s-a format ca disciplină științifică. În această perioadă se dezvoltă metode și algoritmi de bază: tăiere, scanarea rastru, primitive (elemente) grafice, modelele de umbrire, imagini de scene spațiale, trasări tridimensionale, lumini de modelare.
* În anii 1980 programele sunt în curs de dezvoltare mai mult ca o disciplină aplicată. Metodele de aplicare a acesteia se dezvoltă în diverse domenii de activitate umană.
* Metodele anilor 1990 de grafică pe calculator au devenit principalul mijloc de dialog între om și calculator, și rămâne valabil până în prezent.

Grafica digitală contemporană este o disciplină tehnico-științifică destul de dificilă, bine concepută și variată. Unele compartimente, cum ar fi transformările geometrice, modalitățile de a descrie curbe și suprafețe, au fost studiate destul de bine. Unele zone continuă să evolueze: metodele de scanare a rasterului, eliminarea liniilor și suprafețelor ascunse, modelarea și utilizarea culorilor și iluminațiilor, texturarea, crearea efectului de transparență și transluciditate, etc.

Aplicarea graficii pe calculator este formată din următoarele domenii principale:

* Vizualizarea informației
* Proiectare
* Modelare (simulare)
* Interfață grafică pentru utilizatori (GUI)

În prezent cunoașterea elementelor de bază ale graficii pe calculator este necesară inginerilor, oamenilor de știință, artiștilor plastici, designerilor, fotografilor, pictorilor de animație etc.

Odată cu apariția unor noi cerințe s-au dezvoltat rapid multiple aplicații software, devenind mai intuitive, mai structurate pentru utilizare.

Bineinteles, grafica pe calculator este impartita in mai multe categorii dupa cum urmeaza:

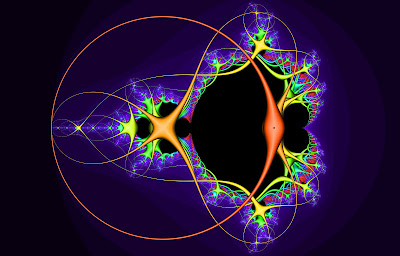
* Grafica bidimensională (*se constituie din 2 algoritm-uri de prezentare – grafică raster și grafică vector*)
* Grafica Fractal
* Grafica tridimensională (*este des utilizată în animație, spoturi publicitare, jocuri la computer etc.*)
* CGI grafica (*CGI – în engleză imagini, personaje generate de computer*)
* Grafica animată
* Grafica video
* Grafica web
* Grafica Multimedia
* Grafica în designul pligrafic

Dar in mod special lucrarea de fata va trata categoria grafici fractale, asa ca sa urmarim cu atentie in continuare.

# Generarea modelelor fractale

## 2.1 Despre modele fractale

**In ziua de azi fractalii generati de computer sunt intalniti peste tot. De la fractal art la articole in cele mai serioase reviste de fizica, interesul pentru aceste structuri neobisnuite este in crestere.**

**Unele dintre aceste forme exista numai in spatii geometrice abstracte, unele exista in natura (brocoli, copaci, corali) iar altele sunt folosite pentru a modela fenomene complexe cum sunt formarea norilor sau modul de functionare al retelei de vase capilare.**

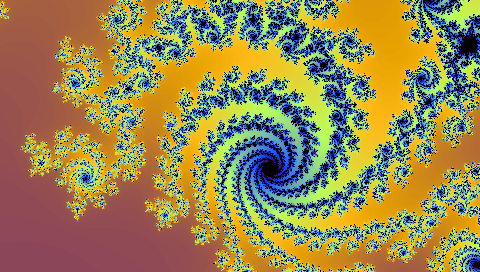
In martie 1980, unitatea centrala ultra-moderna a institutului de cercetare IBM Yorktown Heights din statul New York trimitea instructiuni unui dispozitiv de imprimare Tektronix.

Acesta marca constiincios puncte in locuri neasteptate, iar atunci cand si-a oprit tacanitul, rezultatul semana cu o mana de praf  imprastiata pe foaia de hartie. Lui Benoit Mandelbrot nu ii venea sa creada.

Realiza  importanta evenimentului, dar ce era mai exact? Imaginea semana cu o fotografie in alb si negru abia iesita din baia de developare. Era prima privire aruncata asupra a ceea ce avea sa devina un simbol in lumea fractalilor-multimea Mandelbrot.

Se intampla matematica experimentala prin excelenta, o apropiere catre domeniul in care matematicienii au mese de laborator asa cum au fizicienii si chimistii. Si ei faceau experimente. Se deschideau perspective noi. Era o eliberare din clima arida a succesiunii “definitie, teorema, demonstratie”, desi intoarcerea la rationamentul riguros se va fi intamplat aproape imediat.

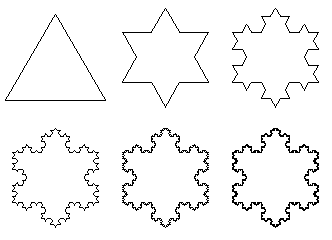
Partea negativa a acestei abordari experimentale era lipsa unui fundament teoretic precedent imaginilor prelucrate. Experimentalistii navigau fara harta.  Mandelbrot a inventat cuvantul “fractal”, dar ce era un fractal de fapt? Exista o definitie precisa cum are in mod normal orice concept matematic?

****La inceput Mandelbrot nu a vrut o astfel de definitie.  Nu a vrut sa distruga magia acestei experiente formuland o definitie eficienta care ar putea fi inadecvata sau ar putea limita. Notiunea de fractal, gandea el, ca si un vin bun, are nevoie de timp inainte de a fi “imbuteliat”.

Fractal art este o forma de arta algoritmica care foloseste fractalii si reprezentarile computerizate pentru a genera imagini, animatii sau  muzica. Artistul britanic William Latham a folosit geometria fractala in operele sale.

Greg Sams a folosit fractalii pentru a crea vederi sau tricouri. Reginald Arkins creaza arta fractala pentru relaxare. Carlos Ginzburg a definit conceptul de “homo fractalus” centrat pe idea ca omul este cel mai reprezentativ exemplu de fractal. Arhitectul spaniol Xavier Vilalta foloseste geometria fractala pentru a aduce inovatii in modul in care se construiesc cladirile eco-friendly.

## 2.2 Curba lui Koch

Faimoasa curba Koch este numita dupa matematicianul suedez Niels Fabian Helge von Koch. Curba fulgul de zapada este practic primul fractal.

Dupa cateva sute de  iteratii lungimea curbei devine mai mare decat diametrul Universului vizibil.

Proprietatea curioasa a curbei Koch este aria finita, aceasta forma aflandu-se la fiecare iteratie in interiorul unui cerc. La fiecare iteratie lungimea curbei creste, este deci o curba ce margineste o arie finita dar are circumferinta infinita!

Iniţial se porneşte cu un segment de dreaptă şi pe segmentul respectiv se ridică un triunghi echilateral, ca şi în figura 1.

|  |  |  |
| --- | --- | --- |
| PIC   |  |  | | --- | --- | | **Figura 1:** | Curba lui Koch după o iteraţie. | |

Pe urmă se repetă procedeul pentru fiecare din cele patru segmente de dreaptă: se ridică un triughi echilateral pe fiecare segment. Rezultatul se poate vedea în figura 2.

|  |
| --- |
| PIC |

**Figura 2:** Curba lui Koch după două iteraţii.

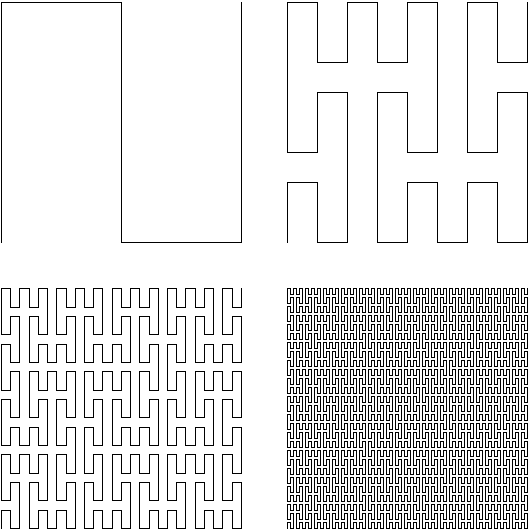
Apoi se repetă acelaşi procedeu, pentru fiecare segment din figura obţinută. Rezultatul este redat în figura 3.

|  |  |  |
| --- | --- | --- |
| PIC   |  |  | | --- | --- | | **Figura 3:** | Curba lui Koch după trei iteraţii. | |

După repetarea de 7 ori a procedeului se obţine o figură de genul celei redate în figura 4.

|  |  |  |
| --- | --- | --- |
| PIC   |  |  | | --- | --- | | **Figura 4:** | Curba lui Koch după şapte iteraţii. | |

## 2.3 Curba lui Peano

Curbele lui Peano mai sunt numite si “curbe care umplu spatiul”. Astfel de curbe sunt create folosind un proces interactiv care produce o curba in zigzag ce acopera intreg spatiul in care se afla.

Un matematician American a folosit curbele Peano pentru a realiza un sistem eficient de distributie pentru o organizatie care livreaza alimente oamenilor saraci si un sistem de distributie al sangelui de catre Crucea Rosie catre spitale.

Mai jos avem un exemplu de curba care umple planul si o curba care umple spatiul.

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

Schema curbei

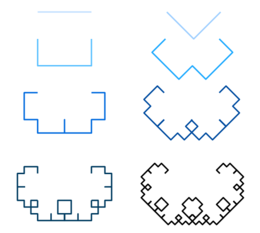
lui Peano

Pentru nivelul n=0 se traseaza un segment de dreapta format din trei parti egale, iar pentru nivelul n=1 se efectueaza urmatoarele trasari in ordinea indicate in schema uramtoare

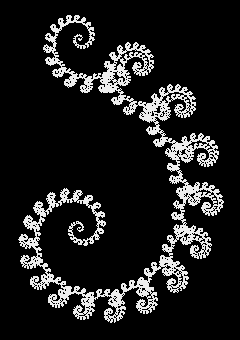
## 2.4 Curba C a lui Lévy



Curba C este o curba fractala care se genereaza astfel: pentru nivelul n, se apeleaza nivelul (n-1), se schimba directia spre dreapta cu , se apeleaza nivelul (n-1) si se schimba directia spre stanga cu



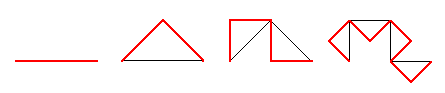
## 2.5 Curba Dragonului



Se traseaza (incepe) cu un segement de dreapta. La prima iteratie, se inlocuieste acest segment cu alte doua segmente, fiecare scalate cu ratia r in asa fel incat segmantul original sa fie ipotenuza unui triunghi isoscel.

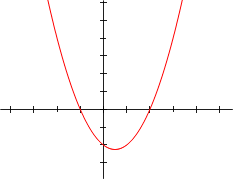
De-alungul segmentului original trasam (punem) cele doua segmente noi spre stanga. La a doua iteratie, inlocuim fiecare dintre segmente cu doua noi segmente la unghi drept, fiecare scalate cu ratia r.

Noile segmente sunt asezate la stanga apoi la dreapta de-alungul segmentelor primei iteratii. Se continua aceasta constructie, alternand mereu segmentele noi intre stanga si dreapta de-alungul segmentelor interatiei anterioare. Acesata constructie genereaza "Curba Dragonului". Figura ce urmeaza arata primele trei iteratii pentru constructia curbei.



# Funcţia de gradul al II-lea

## 3.1 Rezolvare

O funcție algebrică de gradul doi, în matematică, este o funcție polinomială de forma f(x)=ax^2+bx+c \,\!, unde a \ne 0 \,\!.  Graficul unei funcții de gradul doi este o parabolă a cărei axă de simetrie este paralelă cu axa *Oy*.

Expresia ax^2+bx+c din definiția unei funcții pătratice, este un polinom de grad 2 sau funcție polinomială de grad 2, pentru că cel mai mare exponent al variabilei x este 2.

Dacă se pune condiția ca funcția pătratică să fie egală cu zero, atunci va rezulta o ecuație pătratică. Soluțiile acestei ecuații sunt numite rădăcini pătrate ale ecuației, sau puncte de nul ale funcției.

### 3.1.1 Originea Cuvantului

Adjectivul *pătratic* vine de la latinescul *quadratum* care înseamnă pătrat. Termenii de forma*x*2 sunt numiți pătrați în algebră, pentru că reprezintă suprafața unui pătrat cu latura *x*. În general, prefix quadr(i)- se referă la numărul 4. Printre exemple se pot enumera quadrilater și cadan. *Quadratum* este latinescul pentru pătrat, pentru că pătratul are 4 laturi.

### 3.1.2 Radacini

Cele două rădăcini ale ecuației de gradul al doilea 0=ax^2+bx+c\,\!, în care a \ne 0 \,\!sunt:

 x = \frac{-b \pm \sqrt{b^2 - 4 a c}}{2 a}. 

* Fie **\Delta = b^2-4ac \,**
* Dacă **\Delta > 0\,\!**, atunci există două rădăcini distincte pentru că \sqrt{\Delta} este un număr real pozitiv.
* Dacă **\Delta = 0\,\!,** atunci cele două rădăcini sunt egale, pentru că \sqrt{\Delta} este zero.
* Dacă **\Delta < 0\,\!,** atunci cele două rădăcini sunt conjugate complexe, pentru că \sqrt{\Delta} este un număr imaginar.

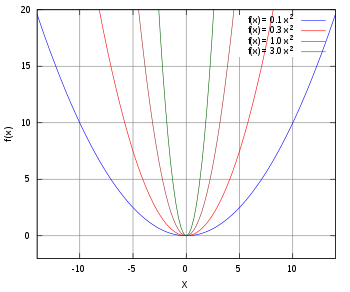
Considerând  r_1 = \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}  și  r_2 = \frac{-b - \sqrt{b^2 - 4 a c}}{2 a}  sau invers, se poate da factor comun  a x^2 + b x + c \,\! sub forma  a(x - r_1)(x - r_2)\,\!.

### 3.1.3 Forme de exprimare a functiilor de gradul al doilea

O funcție de gradul al doilea poate fi exprimată în trei forme principale:

* f(x) = a x^2 + b x + c \,\! se numește *formă dezvoltată*,
* f(x) = a(x - x_1)(x - x_2)\,\! se numește *forma factorizată*, în care  x_1  și  x_2  sunt rădăcinile ecuației
* f(x) = a(x - h)^2 + k \,\! este forma canonică, în care *h* și *k* reprezintă abscisa, respectiv ordonata punctului de extrem.

## 3.2 Reprezentare grafica

Indiferent de forma în care este exprimată ea, graficul unei funcții de gradul al doilea este o parabolă.

* Dacă a > 0 \,\!, parabola are deschiderea în sus.
* Dacă a < 0 \,\!, parabola are deschiderea în jos.

Coeficientul *a* controlează viteza de creștere (sau descreștere) a funcției de la vârf, un *a* pozitiv mai mare făcând ca funcția să crească mai rapid și ca graficul să pară mai strâns.

Coeficienții *b* și *a* împreună controlează axa de simetrie a parabolei (precum și abscisa vârfului) care este x = -\frac{b}{2a}.

Coeficientul *b* singur este înclinația parabolei la intersecția cu axa Oy.

Coeficientul *c* controlează înălțimea parabolei, adică locul în care ea intersectează axa Oy.

Vârful unei parabole este punctul în care ea atinge maximul sau minimul, fiind astfel *punctul de extrem*. Dacă funcția este scrisă în formă canonică, vârful este (h, k)\,\!.

*Forma generală*

f(x) = a x^2 + b x + c \,\!

se poate transforma în

 f(x) = a\left(x + \frac{b}{2a}\right)^2 - \frac{b^2-4ac}{4 a} ,

și deci vârful parabolei are coordonatele

 \left(-\frac{b}{2a}, -\frac{\Delta}{4 a}\right). 

Dacă ecuația este în forma factorizată

f(x) = a(x - r_1)(x - r_2) \,\!

media celor două rădăcini,

\frac{r_1 + r_2}{2} \,\!

este abscisa vârfului, care are, deci, coordonatele

 \left(\frac{r_1 + r_2}{2}, f(\frac{r_1 + r_2}{2})\right).\!

Vârful este punct de maxim dacă a < 0 \,\! și punct de minim dacă a > 0 \,\!.

Dreapta verticală care trece prin vârf este axa de simetrie a parabolei.

 x=h=-\frac{b}{2a} 

*Puncte de maxim și de minim*

În analiza matematică, coordonatele vârfului, ca punct de extrem al funcției, se pot obține aflând rădăcina derivatei:

f(x)=ax^2+bx+c \quad \Rightarrow \quad f'(x)=2ax+b \,\!,

ceea ce dă

x=-\frac{b}{2a}

cu valoarea corespunzătoare

f(x) = a \left (-\frac{b}{2a} \right)^2+b \left (-\frac{b}{2a} \right)+c = -\frac{(b^2-4ac)}{4a} = -\frac{\Delta}{4a} \,\!,

 \left (-\frac {b}{2a}, -\frac {\Delta}{4a} \right). și deci coordonatele vârfului pot fi exprimate:

# Jocul trandafirului

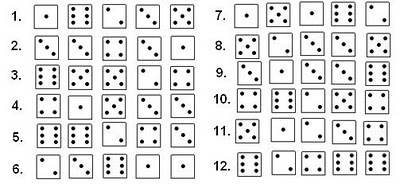
## 4.1 Scurt istoric

Acest joc a devenit celebru printre cercurile de programatori inca din anul 1970. Petale din jurul trandafirul este un puzzle matematic in care principalul obiect de lucru este o formula din care rezulta numarul petalelor din jurul trandafirului reprezentat prin cele 5 fete ale zarurilor.

## 4.2 Cum se joaca

Regulile jocului sunt simple si usoare dupa cum urmeaza:

1. Numele jocului este semnificativ pentru rezolvarea puzzle-ului;
2. Raspunsul este intodeauna un numar par sau zero;
3. Odata descoperit rationamentul de rezolvare, acesta trebuie pastrat secret.



# Bibliografie

1. Totul despre C si C++ - Dr. Kris Jamsa Lars Klander, Teora, 2005
2. Grafică pe calculator în limbajele PASCAL şi C. Implementare şi aplicatii, EDITURA TEHNICĂ (1992) , vol. I , Implementare, 209 pag. ( în colaborare cu A.Posea, C. Constantinescu, I. Nistor). (ISBN 973-31-0406-X) [carte citată de diverşi specialişti]
3. Grafică pe calculator în limbajele PASCAL şi C. Implementare şi aplicatii, EDITURA TEHNICĂ (1992), vol II, Aplicatii, 249 pag.
4. http://ro.wikipedia.org/wiki/Fractal
5. http://ro.wikipedia.org/wiki/Func%C8%9Bie\_algebric%C4%83\_de\_gradul\_doi
6. http://ro.wikipedia.org/wiki/Grafic%C4%83\_digital%C4%83
7. http://en.wikipedia.org/wiki/Petals\_Around\_the\_Rose

# Anexa - codul sursă al aplicaţiei

aplicatie\_atestat.cpp – 20 KB

#include<iostream.h>

#include<math.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

#include<dos.h>

#include<stdio.h>

#include<string.h>

int x, y, x1, y1;

float alfa;

void stanga(float unghi)

{alfa+=unghi\*M\_PI/180;}

void dreapta(float unghi)

{alfa-=unghi\*M\_PI/180;}

void C(float n, float L)

{if(n==0)

linerel(int(L\*cos(alfa)),int(L\*sin(alfa)));

else

{C(n-1,L); dreapta(90);

C(n-1,L); stanga(90);

}

}

void Dragon(int n, float L, int v)

{if(n==0)

linerel(int(L\*cos(alfa)),int(L\*sin(alfa)));

else

{dreapta(45\*v);

Dragon(n-1,L,1);

stanga(90\*v);

Dragon(n-1,L,-1);

dreapta(45\*v);

}

}

void Koch(int n, float L)

{if(n==0)

linerel(int(L\*cos(alfa)),int(L\*sin(alfa)));

else

{Koch(n-1,L/3); stanga(60);

Koch(n-1,L/3); dreapta(120);

Koch(n-1,L/3); stanga(60);

Koch(n-1,L/3);

}

}

void nucleu\_Koch(int n, float L)

{Koch(n,L); dreapta(120);

Koch(n,L); dreapta(120);

Koch(n,L); dreapta(120);

}

void PEANO(int n, float L)

{if(n==0)

linerel(int(L\*cos(alfa)),int(L\*sin(alfa)));

else

{PEANO(n-1,L/3); stanga(90);

PEANO(n-1,L/3); dreapta(90);

PEANO(n-1,L/3); dreapta(90);

PEANO(n-1,L/3); dreapta(90);

PEANO(n-1,L/3); stanga(90);

PEANO(n-1,L/3); stanga(90);

PEANO(n-1,L/3); stanga(90);

PEANO(n-1,L/3); dreapta(90);

PEANO(n-1,L/3);

}

}

void curba\_c()

{int n, L, errorcode, i;

char c[25], sw;

do{

cout<<"Dimensiunea desenului ";

cin>>L;

cout<<"Factorul de scara ";

cin>>n;

cout<<"Cum doriti sa colorez desenul?\n";

cout<<"- albastru\n- galben\n- rosu\n- verde\n";

gets(c);

if(strcmp(c,"albastru")==0)

i=1;

else

if(strcmp(c,"verde")==0)

i=2;

else

if(strcmp(c,"rosu")==0)

i=4;

else

if(strcmp(c,"galben")==0)

i=14;

else

i=15;

cleardevice();

gotoxy(10,2);

cout<<"Curba C a lui Levy de dimensiune "<<L<<" si factor de scara "<<n<<" este:";

moveto(100,300);

setcolor(i);

C(n,L);

gotoxy(20,25);

cout<<"Doriti sa realizati un nou desen? (D/N) ";

cin>>sw;

cleardevice();

gotoxy(1,1);

}

while(sw=='D' || sw=='d');

}

void dragon()

{int n, L, i;

char x[25], sw;

do{

cout<<"Varsta dragonului este ";

cin>>n;

cout<<"Factorul de scara este ";

cin>>L;

cout<<"Cum doriti sa-l colorati?\n";

cout<<"- albastru\n- galben\n- rosu\n- verde\n";

gets(x);

if(strcmp(x,"albastru")==0)

i=1;

else

if(strcmp(x,"verde")==0)

i=2;

else

if(strcmp(x,"rosu")==0)

i=4;

else

if(strcmp(x,"galben")==0)

i=14;

else

i=15;

cleardevice();

moveto(200,250);

setcolor(15);

gotoxy(12,2);

cout<<"Dragonul cu varsta de "<<n<<" ani la un factor de scara "<<L<<" este:";

setcolor(i);

Dragon(n,L/sqrt(2),1);

gotoxy(15,25);

cout<<"Doriti sa dati nastere unui nou dragon? (D/N) ";

cin>>sw;

cleardevice();

gotoxy(1,1);

}while(sw=='D' || sw=='d');

}

void koch()

{int n, L, i;

char c[25], sw;

cleardevice();

do{

cout<<"Nivelul ";

cin>>L;

cout<<"Dimensiunea laturi ";

cin>>n;

cout<<"Cum doriti sa colorez desenul?\n";

cout<<"- albastru\n- galben\n- rosu\n- verde\n";

gets(c);

if(strcmp(c,"albastru")==0)

i=1;

else

if(strcmp(c,"verde")==0)

i=2;

else

if(strcmp(c,"rosu")==0)

i=4;

else

if(strcmp(c,"galben")==0)

i=14;

else

i=15;

cleardevice();

gotoxy(17,2);

cout<<"Fulgul de nea de nivel "<<L<<" si latura "<<n<<" este:";

moveto(170,300);

setcolor(i);

nucleu\_Koch(n,L);

gotoxy(15,25);

cout<<"Doriti sa realizati un nou desen? (D/N) ";

cin>>sw;

cleardevice();

gotoxy(1,1);

}while(sw=='d' || sw=='D');

}

void peano()

{int n, L, i;

char c[25], sw;

cleardevice();

do{

cout<<"Nivel ";

cin>>n;

cout<<"Dimensiune ";

cin>>L;

cout<<"Cum doriti sa colorez desenul?\n";

cout<<"- albastru\n- galben\n- rosu\n- verde\n ";

gets(c);

if(strcmp(c,"albastru")==0)

i=1;

else

if(strcmp(c,"verde")==0)

i=2;

else

if(strcmp(c,"rosu")==0)

i=4;

else

if(strcmp(c,"galben")==0)

i=14;

else

i=15;

cleardevice();

gotoxy(15,2);

cout<<"Curba lui Peano de nivel "<<n<<" si dimensiune "<<L<<" este:";

setcolor(i);

moveto(200,200);

PEANO(n,L);

gotoxy(15,25);

cout<<"Doriti sa realizati un nou desen? (D/N) ";

cin>>sw;

cleardevice();

gotoxy(1,1);

}while(sw=='D' || sw=='d');

}

void functie\_gr2()

{ int prev\_x, prev\_y, a, b, c;

float end\_x, end\_y, x, y, x\_varf, y\_varf, x\_varf1, y\_varf1, x1, x2, xp, yp, delta;

char sw, sw2;

do{

prev\_x=0; prev\_y=0;

end\_x=639; end\_y=479;

x=10; y=10;

gotoxy(20,2);

cout<<"Functia de gradul al II-lea\n\n";

cout<<"a="; cin>>a;

cout<<"b="; cin>>b;

cout<<"c="; cin>>c; cout<<"\n";

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

x\_varf=((-b)/2\*a);

y\_varf=((-delta)/4\*a);

x\_varf1=x\_varf\*30;

y\_varf1=y\_varf\*30;

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

yp=((-delta)/4\*a);

if(delta>=0)

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

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

cout<<"Functia introdusa este f(x) = "<<a<<"x^2 + "<<b<<"x + "<<c<<"\n\n";

if(delta==0)

cout<<"Ecuatia are o singura solutie reala x = "<<x1;

else

if(delta>0)

cout<<"Ecuatia are doua solutii reale distincte x1 = "<<x1<<" si x2 = "<<x2;

else

if(delta<0)

cout<<"Ecuatia nu are solutii reale";

cout<<"\nVarful parabolei are coordonatele V("<<(-b/2\*a)<<","<<(-delta/4\*a)<<")";

cout<<"\n\n Tabelul de variatie este";

gotoxy(3,15);

cout<<"x";

gotoxy(1,17);

cout<<"f(x)";

if(delta==0)

{

gotoxy(20,15);

cout<<xp;

gotoxy(30,15);

cout<<x1;

if(a<0)

{gotoxy(8,17);

cout<<"- - - - - - - - - - - - - - - - -";}

else

{gotoxy(8,17);

cout<<"+ + + + + + + + + + + + + + + + +";}

gotoxy(20,17);

cout<<yp;

gotoxy(30,17);

cout<<"0";

}

if(delta>0)

{if((x1<x2)){

gotoxy(15,15);

cout<<x1;

gotoxy(25,15);

cout<<xp;

gotoxy(35,15);

cout<<x2;

if(a>0)

{gotoxy(8,17);

cout<<"+ + + - - - - - - - - - - + + +";}

else

{gotoxy(8,17);

cout<<"- - - + + + + + + + + + + - - -";}

gotoxy(15,17);

cout<<"0";

gotoxy(25,17);

cout<<yp;

gotoxy(35,17);

cout<<"0";}

if(x1>x2){

gotoxy(15,15);

cout<<x2;

gotoxy(25,15);

cout<<xp;

gotoxy(35,15);

cout<<x1;

if(a>0)

{gotoxy(8,17);

cout<<"+ + + - - - - - - - - - - + + +";}

else

{gotoxy(8,17);

cout<<"- - - - - - - + + + + + + - - -";}

gotoxy(15,17);

cout<<"0";

gotoxy(25,17);

cout<<yp;

gotoxy(35,17);

cout<<"0";}

}

if(delta<0)

{gotoxy(23,15);

cout<<xp;

if(a>=0)

{gotoxy(8,17);

cout<<" + + + + + + + + + + + + + + +";}

else

{gotoxy(8,17);

cout<<" - - - - - - - - - - - - - -";}

gotoxy(23,17);

cout<<yp;}

line(1,245,350,245);

line(40,220,40,280);

gotoxy(1,20);

cout<<"Doriti sa afiesez graficul functiei introduse? (D/N) ";

cin>>sw;

if(sw=='D' || sw=='d'){

cleardevice();

gotoxy(1,1);

cout<<"Graficul functiei\nf(x) = "<<a<<"x^2 + "<<b<<"x + "<<c;

delay(2000);

if((end\_y/2+y\_varf1)<=end\_x)

line(x,end\_y/2+(y\_varf1),end\_x,end\_y/2+(y\_varf1));

if((end\_x/2-x\_varf1)<=end\_y)

line(end\_x/2-(x\_varf1),y,end\_x/2-(x\_varf1),end\_y);

if (a>0) {

for (x=-1\*(int)(end\_x/2); x<(int)(end\_x/2); x+=.001)

{

// valori parabola

y = .03\*(x\*x);

//putpixel((int)(end\_x-x) - (int)(end\_x/2.0), (int)(end\_y-y) - (int)(end\_y/2.0), WHITE);

line((int)(end\_x-x) - (int)(end\_x/2.0), (int)(end\_y-y) - (int)(end\_y/2.0), prev\_x, prev\_y);

//cand folosim line() trebuie sa retinem valorile anterioare x,y

prev\_x = (int)(end\_x - x) - (int)(end\_x/2.0);

prev\_y = (int)(end\_y - y) - (int)(end\_y/2.0);

}

}

else if (a<0) {

for (x=-1\*(int)(end\_x/2); x<(int)(end\_x/2); x+=.001)

{y = .03\*(x\*x);

putpixel((int)(end\_x+x) - (int)(end\_x/2.0), (int)(end\_y+y) - (int)(end\_y/2.0), WHITE);

}}

}

gotoxy(1,25);

cout<<"Doriti sa introduceti\no alta functie? (D/N) ";

cin>>sw2;

cleardevice();

}while(sw2=='d' || sw2=='D');

}

void joc()

{ int zar\_1, zar\_2, zar\_3, zar\_4, zar\_5, nr\_petale, petale, k, scor\_total=0;

char cuv, inceput;

setbkcolor(BLUE);

gotoxy(27,2);

cout<<"Doriti sa incepeti jocul? (D/N) ";

cin>>inceput;

if(inceput=='d' || inceput=='D')

{

while(inceput=='d' || inceput=='D')

{nr\_petale=0;

rectangle(20,40,123,143);

rectangle(143,40,246,143);

rectangle(266,40,369,143);

rectangle(389,40,492,143);

rectangle(512,40,615,143);

gotoxy(20,11);

cout<<"Indicatii:";

gotoxy(20,12);

cout<<"- numele jocului ARE legatura cu rezolvarea lui;";

gotoxy(20,13);

cout<<"- rezultatul poate fi 0 sau un numar par;";

gotoxy(20,14);

cout<<"- puteti afla rezultatul corect atunci cand doriti.";

/\* initial,ize random seed: \*/

srand (time(NULL));

/\* generate random number between 1 and 6: \*/

zar\_1 = rand() % 6 + 1;

if(zar\_1%2!=0)

nr\_petale+=zar\_1-1;

zar\_2 = rand() % 6 + 1;

if(zar\_2%2!=0)

nr\_petale+=zar\_2-1;

zar\_3 = rand() % 6 + 1;

if(zar\_3%2!=0)

nr\_petale+=zar\_3-1;

zar\_4 = rand() % 6 + 1;

if(zar\_4%2!=0)

nr\_petale+=zar\_4-1;

zar\_5 = rand() % 6 + 1;

if(zar\_5%2!=0)

nr\_petale+=zar\_5-1;

switch (zar\_1)

{

case 1: {// - Zarul 1 - Valoarea 1 - //

circle(70,90,7);

floodfill(70,90,WHITE);

} break;

case 2: {// - Zarul 1 - Valoarea 2 - //

circle(40,123,7);

floodfill(40,123,WHITE);

circle(103,60,7);

floodfill(103,60,WHITE);

} break;

case 3: {// - Zarul 1 - Valoarea 3 - //

circle(40,123,7);

floodfill(40,123,WHITE);

circle(70,90,7);

floodfill(70,90,WHITE);

circle(103,60,7);

floodfill(103,60,WHITE);

} break;

case 4: {// - Zarul 1 - Valoarea 4 - //

circle(40,123,7);

floodfill(40,123,WHITE);

circle(40,60,7);

floodfill(40,60,WHITE);

circle(103,60,7);

floodfill(103,60,WHITE);

circle(103,123,7);

floodfill(103,123,WHITE);

} break;

case 5: {// - Zarul 1 - Valoarea 5 - //

circle(40,123,7);

floodfill(40,123,WHITE);

circle(40,60,7);

floodfill(40,60,WHITE);

circle(70,90,7);

floodfill(70,90,WHITE);

circle(103,60,7);

floodfill(103,60,WHITE);

circle(103,123,7);

floodfill(103,123,WHITE);

} break;

case 6: {// - Zarul 1 - Valoarea 6 - //

circle(40,60,7);

floodfill(40,60,WHITE);

circle(103,123,7);

floodfill(103,123,WHITE);

circle(103,60,7);

floodfill(103,60,WHITE);

circle(40,123,7);

floodfill(40,123,WHITE);

circle(103,90,7);

floodfill(103,90,WHITE);

circle(40,90,7);

floodfill(40,90,WHITE);

} break;

}

switch (zar\_2)

{

case 1: {// - Zarul 2 - Valoarea 1 - //

circle(193,90,7);

floodfill(193,90,WHITE);

} break;

case 2: {// - Zarul 2 - Valoarea 2 - //

circle(163,123,7);

floodfill(163,123,WHITE);

circle(226,60,7);

floodfill(226,60,WHITE);

} break;

case 3: {// - Zarul 2 - Valoarea 3 - //

circle(163,123,7);

floodfill(163,123,WHITE);

circle(193,90,7);

floodfill(193,90,WHITE);

circle(226,60,7);

floodfill(226,60,WHITE);

} break;

case 4: {// - Zarul 2 - Valoarea 4 - //

circle(163,123,7);

floodfill(163,123,WHITE);

circle(226,60,7);

floodfill(226,60,WHITE);

circle(226,123,7);

floodfill(226,123,WHITE);

circle(163,60,7);

floodfill(163,60,WHITE);

} break;

case 5: {// - Zarul 2 - Valoarea 5 - //

circle(163,123,7);

floodfill(163,123,WHITE);

circle(226,60,7);

floodfill(226,60,WHITE);

circle(193,90,7);

floodfill(193,90,WHITE);

circle(226,123,7);

floodfill(226,123,WHITE);

circle(163,60,7);

floodfill(163,60,WHITE);

} break;

case 6: {// - Zarul 2 - Valoarea 6 - //

circle(163,123,7);

floodfill(163,123,WHITE);

circle(226,60,7);

floodfill(226,60,WHITE);

circle(163,90,7);

floodfill(163,90,WHITE);

circle(226,90,7);

floodfill(226,90,WHITE);

circle(226,123,7);

floodfill(226,123,WHITE);

circle(163,60,7);

floodfill(163,60,WHITE);

} break;

}

switch (zar\_3)

{

case 1: {// - Zarul 3 - Valoarea 1 - //

circle(316,90,7);

floodfill(316,90,WHITE);

} break;

case 2: {// - Zarul 3 - Valoarea 2 - //

circle(286,123,7);

floodfill(286,123,WHITE);

circle(349,60,7);

floodfill(349,60,WHITE);

} break;

case 3: {// - Zarul 3 - Valoarea 3 - //

circle(286,123,7);

floodfill(286,123,WHITE);

circle(316,90,7);

floodfill(316,90,WHITE);

circle(349,60,7);

floodfill(349,60,WHITE);

} break;

case 4: {// - Zarul 3 - Valoarea 4 - //

circle(286,123,7);

floodfill(286,123,WHITE);

circle(286,60,7);

floodfill(286,60,WHITE);

circle(349,60,7);

floodfill(349,60,WHITE);

circle(349,123,7);

floodfill(349,123,WHITE);

} break;

case 5: {// - Zarul 3 - Valoarea 5 - //

circle(286,123,7);

floodfill(286,123,WHITE);

circle(286,60,7);

floodfill(286,60,WHITE);

circle(316,90,7);

floodfill(316,90,WHITE);

circle(349,60,7);

floodfill(349,60,WHITE);

circle(349,123,7);

floodfill(349,123,WHITE);

} break;

case 6: {// - Zarul 3 - Valoarea 6 - //

circle(286,123,7);

floodfill(286,123,WHITE);

circle(286,60,7);

floodfill(286,60,WHITE);

circle(286,90,7);

floodfill(286,90,WHITE);

circle(349,60,7);

floodfill(349,60,WHITE);

circle(349,123,7);

floodfill(349,123,WHITE);

circle(349,90,7);

floodfill(349,90,WHITE);

} break;

}

switch (zar\_4)

{

case 1: {// - Zarul 4 - Valoarea 1 - //

circle(439,90,7);

floodfill(439,90,WHITE);

} break;

case 2: {// - Zarul 4 - Valoarea 2 - //

circle(409,123,7);

floodfill(409,123,WHITE);

circle(472,60,7);

floodfill(472,60,WHITE);

} break;

case 3: {// - Zarul 4 - Valoarea 3 - //

circle(409,123,7);

floodfill(409,123,WHITE);

circle(439,90,7);

floodfill(439,90,WHITE);

circle(472,60,7);

floodfill(472,60,WHITE);

} break;

case 4: {// - Zarul 4 - Valoarea 4 - //

circle(409,123,7);

floodfill(409,123,WHITE);

circle(409,60,7);

floodfill(409,60,WHITE);

circle(472,60,7);

floodfill(472,60,WHITE);

circle(472,123,7);

floodfill(472,123,WHITE);

} break;

case 5: {// - Zarul 4 - Valoarea 5 - //

circle(409,123,7);

floodfill(409,123,WHITE);

circle(409,60,7);

floodfill(409,60,WHITE);

circle(439,90,7);

floodfill(439,90,WHITE);

circle(472,60,7);

floodfill(472,60,WHITE);

circle(472,123,7);

floodfill(472,123,WHITE);

} break;

case 6: {// - Zarul 4 - Valoarea 6 - //

circle(409,123,7);

floodfill(409,123,WHITE);

circle(409,60,7);

floodfill(409,60,WHITE);

circle(409,90,7);

floodfill(409,90,WHITE);

circle(472,60,7);

floodfill(472,60,WHITE);

circle(472,123,7);

floodfill(472,123,WHITE);

circle(472,90,7);

floodfill(472,90,WHITE);

} break;

}

switch (zar\_5)

{

case 1: {// - Zarul 5 - Valoarea 1 - //

circle(562,90,7);

floodfill(562,90,WHITE);

} break;

case 2: {// - Zarul 5 - Valoarea 2 - //

circle(532,123,7);

floodfill(532,123,WHITE);

circle(595,60,7);

floodfill(595,60,WHITE);

} break;

case 3: {// - Zarul 5 - Valoarea 3 - //

circle(532,123,7);

floodfill(532,123,WHITE);

circle(562,90,7);

floodfill(562,90,WHITE);

circle(595,60,7);

floodfill(595,60,WHITE);

} break;

case 4: {// - Zarul 5 - Valoarea 4 - //

circle(532,123,7);

floodfill(532,123,WHITE);

circle(532,60,7);

floodfill(532,60,WHITE);

circle(595,60,7);

floodfill(595,60,WHITE);

circle(595,123,7);

floodfill(595,123,WHITE);

} break;

case 5: {// - Zarul 5 - Valoarea 5 - //

circle(532,123,7);

floodfill(532,123,WHITE);

circle(532,60,7);

floodfill(532,60,WHITE);

circle(562,90,7);

floodfill(562,90,WHITE);

circle(595,60,7);

floodfill(595,60,WHITE);

circle(595,123,7);

floodfill(595,123,WHITE);

} break;

case 6: {// - Zarul 5 - Valoarea 6 - //

circle(532,123,7);

floodfill(532,123,WHITE);

circle(532,90,7);

floodfill(532,90,WHITE);

circle(532,60,7);

floodfill(532,60,WHITE);

circle(595,60,7);

floodfill(595,60,WHITE);

circle(595,90,7);

floodfill(595,90,WHITE);

circle(595,123,7);

floodfill(595,123,WHITE);

} break;

}

gotoxy(1,15);

cout<<"\nCat de multe petale sunt in jurul trandafirului?";

cout<<"\nDoriti sa afisez rezultatul pentru a va orienta? (D/N) ";

cin>>cuv;

if(cuv=='d' || cuv=='D')

k=0;

else

k=1;

if(k==0)

cout<<"\nNumarul petalelor este "<<nr\_petale;

if(k){

cout<<"\nIntoduceti numarul de petale din jurul trandafirului\nNumar petale=";

cin>>petale;

if(petale==nr\_petale)

{cout<<"\nCORECT! Numarul petalelor este "<<nr\_petale;

scor\_total++;}

else

{cout<<"\nGRESIT! Numarul petalelor este "<<nr\_petale;

scor\_total--;}

}

cout<<"\nDoriti sa continuati jocul (D/N)? ";

cin>>inceput;

cleardevice();

gotoxy(30,2);

cout<<"Petalele Trandafirului";

}

}

else

cout<<"\n INSPIRATIE! "<<'\2';

}

void main(){

int i, errorcode, optiune;

int gdriver = DETECT, gmode;

initgraph(&gdriver,&gmode,"C:\\BORLANDC\\BGI");

errorcode=graphresult();

while(errorcode)

{cout<<"Eroare la placa grafica. Verificati fisierele BGI";

getch();

exit(1);

}

char ATESTAT[]="ATESTAT INFORMATICA", AUTOR[]="ISPAS ADRIAN, XII-B", OPTIUNE[]="Optiunea dvs: ",

OPT1[]="1. Curba lui Koch", OPT2[]="2. Curba lui Peano", OPT3[]="3. Curba C", OPT4[]="4. Curba dragonului", OPT5[]="5. Functia de gradul II", OPT6[]="6. Joc", OPT7[]="7. Inchide aplicatia";

do{

setbkcolor(RED);

rectangle(0,1,getmaxx(),getmaxy());

setfillstyle(SOLID\_FILL,0);

bar(165,115,494,344);

setfillstyle(SOLID\_FILL,BLUE);

bar(150,100,479,329);

rectangle(149,99,480,330);

outtextxy(235, 110, ATESTAT);

outtextxy(325, 315, AUTOR);

outtextxy(165, 140, OPT1);

outtextxy(165, 160, OPT2);

outtextxy(165, 180, OPT3);

outtextxy(165, 200, OPT4);

outtextxy(165, 220, OPT5);

outtextxy(165, 240, OPT6);

outtextxy(165, 260, OPT7);

outtextxy(165, 292, OPTIUNE);

gotoxy(35,19);

cin>>optiune;

switch(optiune){

case 1: {cleardevice();

gotoxy(1,1);

setbkcolor(BLACK);

koch();

i=1;

setcolor(WHITE);

break;}

case 2: {cleardevice();

gotoxy(1,1);

setbkcolor(BLACK);

peano();

i=1;

setcolor(WHITE);

break;}

case 3: {cleardevice();

gotoxy(1,1);

setbkcolor(BLACK);

curba\_c();

i=1;

setcolor(WHITE);

break;}

case 4: {cleardevice();

gotoxy(1,1);

setbkcolor(BLACK);

dragon();

i=1;

setcolor(WHITE);

break;}

case 5: {cleardevice();

gotoxy(1,1);

setbkcolor(BLACK);

functie\_gr2();

i=1;

setcolor(WHITE);

break;}

case 6: {cleardevice();

gotoxy(1,1);

setbkcolor(BLACK);

joc();

i=1;

setcolor(WHITE);

cleardevice();

break;}

case 7: {i=0; exit(1);}

break;

}

}while(i==1);

getch();

}