**Flood Fill**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

void flood(int x,int y,int new\_color,int old\_color){

if(getpixel(x,y)==old\_color){

delay(1);

putpixel(x,y,new\_color);

flood(x+1,y,new\_color,old\_color);

flood(x-1,y,new\_color,old\_color);

flood(x,y+1,new\_color,old\_color);

flood(x,y-1,new\_color,old\_color);

}

}

int main(){

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

rectangle(50,50,100,100);

flood(51,51,10,0);

getch();

closegraph();

return 0;

}

**DDA**

#include <stdio.h>

#include <conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

int x1, y1, x2, y2,x, y, i, steps;

float dx,dy,xinc,yinc;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("Enter Co-ordinates x1 y1 x2 y2 : ");

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

dx = (float) x2 - x1 ;

dy = (float) y2 - y1 ;

if(dx > dy){

steps =(int) dx;

}else{

steps =(int) dy;

}

xinc = dx /steps ;

yinc = dy /steps ;

x = x1;

y = y1;

i = 1;

while(i<=steps){

putpixel(x,y,WHITE);

x += xinc;

y += yinc;

i++;

}

getch();

closegraph();

}

**2D Use Input 100 100 150 100 125 125 tx = 100 ty = 100 sx = 2 sy = 2 angle = 45**

#include <stdio.h>

#include <conio.h>

#include<graphics.h>

#include <math.h>

void draw(int x1,int y1,int x2,int y2,int x3,int y3){

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

}

void main()

{

int gd=DETECT,gm;

int x1, y1, x2, y2, x3, y3, tx, ty, sx, sy, angle;

int nx1, ny1, nx2, ny2, nx3, ny3;

float t;

int choice;

choice = 1;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("Enter Co-ordinates of Triangle : ");

scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2,&x3,&y3);

nx1 = x1 ,ny1 = y1 , nx2 = x2, ny2 = y2, nx3 = x3, ny3 = y3;

draw(x1, y1, x2, y2, x3, y3);

while (choice < 4 && choice > 0)

{

printf("1.Translation \t 2.Rotation \t 3.Scaling \t 4.Exit\n");

printf("Enter your choice : ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("\nEnter Translating Factors tx and ty : ");

scanf("%d%d", &tx,&ty);

nx1 = tx+nx1;

ny1 = ty+ny1;

nx2 = tx+nx2;

ny2 = ty+ny2;

nx3 = tx+nx3;

ny3 = ty+ny3;

draw(nx1, ny1, nx2, ny2, nx3, ny3);

getch();

break;

case 2:

printf("\nEnter Rotating angle : ");

scanf("%d",&angle);

t = 3.14 \* angle / 180;

nx1 = abs(nx1\*cos(t) - ny1\*sin(t));

ny1 = abs(nx1\*sin(t) + ny1\*cos(t));

nx2 = abs(nx2\*cos(t) - ny2\*sin(t));

ny2 = abs(nx2\*sin(t) + ny2\*cos(t));

nx3 = abs(nx3\*cos(t) - ny3\*sin(t));

ny3 = abs(nx3\*sin(t) + ny3\*cos(t));

draw(nx1,ny1,nx2,ny2,nx3,ny3);

getch();

break;

case 3:

printf("\nEnter Scaling Factor sx and sy : ");

scanf("%d%d",&sx,&sy);

nx1 = sx\*nx1;

ny1 = sy\*ny1;

nx2 = sx\*nx2;

ny2 = sy\*ny2;

nx3 = sx\*nx3;

ny3 = sy\*ny3;

draw(nx1, ny1, nx2, ny2, nx3, ny3);

getch();

break;

}

closegraph();

}

**3D Use Input 100 100 150 100 tx = 100 ty = 100 sx = 2 sy = 2 angle = 45**

#include <stdio.h>

#include <conio.h>

#include<graphics.h>

#include <math.h>

void main()

{

int gd=DETECT,gm;

int x1, y1, x2, y2, tx, ty, sx, sy, angle;

int nx1, ny1, nx2, ny2,depth;

float t;

int choice;

choice = 1;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("Enter Co-ordinates x1 y1 x2 y2 : ");

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

nx1 = x1;

ny1 = y1;

nx2 = x2;

ny2 = y2;

depth = (nx2 - nx1)/4;

bar3d(x1, y1, x2, y2,depth,2);

while (choice < 4 && choice > 0)

{

printf("1.Translation \t 2.Rotation \t 3.Scaling \t 4.Exit\n");

printf("Enter your choice : ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("\nEnter Translating Factors tx and ty : ");

scanf("%d%d", &tx,&ty);

nx1 = tx+nx1;

ny1 = ty+ny1;

nx2 = tx+nx2;

ny2 = ty+ny2;

depth = (nx2 - nx1)/4;

bar3d(nx1, ny1, nx2, ny2,depth,1);

getch();

break;

case 2:

printf("\nEnter Rotating angle : ");

scanf("%d",&angle);

t = 3.14 \* angle / 180;

nx1 = abs(nx1\*cos(t) - ny1\*sin(t));

ny1 = abs(nx1\*sin(t) + ny1\*cos(t));

nx2 = abs(nx2\*cos(t) - ny2\*sin(t));

ny2 = abs(nx2\*sin(t) + ny2\*cos(t));

depth = (nx2 - nx1)/4;

bar3d(nx1,ny1,nx2,ny2,depth,2);

getch();

break;

case 3:

printf("\nEnter Scaling Factor sx and sy : ");

scanf("%d%d",&sx,&sy);

nx1 = sx\*nx1;

ny1 = sy\*ny1;

nx2 = sx\*nx2;

ny2 = sy\*ny2;

depth = (nx2 - nx1)/4;

bar3d(nx1, ny1, nx2, ny2,depth,1);

getch();

break;

default:

break;

}

}

closegraph();

}

**Animation**

#include <stdio.h>

#include <conio.h>

#include<graphics.h>

#include<dos.h>

void main()

{

   int gd=DETECT,gm;

   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

   int x1,y1,x2,y2,i;

   x1=100,y1=100,x2=200,y2=200;

   rectangle(x1,y1,x2,y2);

   for(i = 1; i <= 350 ; i++){

      delay(2);

      cleardevice();

      rectangle(x1+i,y1,x2+i,y2);

   }

   getch();

   closegraph();

}

**Bezier Curve : Use 200 300 300 400 300 300 100 200**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void main(){

int gd=DETECT,gm;

int x[4],y[4],i;

float put\_x,put\_y, t;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("Enter The 4 control points : ");

scanf("%d%d%d%d%d%d%d%d",&x[0],&y[0],&x[1],&y[1],&x[2],&y[2],&x[3],&y[3]);

for(i = 0; i< 4; i++){

putpixel(x[i],y[i],BLUE);

}

for(t = 0.0; t<1.0; t+=0.001){

put\_x = pow(1-t,3)\*x[0]+ 3\*t\* pow(1-t,2) \* x[1] + 3\* pow(t,2) \*(1-t)\*x[2] + pow(t,3)\*x[3];

put\_y = pow(1-t,3)\*y[0]+ 3\*t\* pow(1-t,2) \* y[1] + 3\* pow(t,2) \*(1-1)\*y[3] + pow(t,3)\*y[3];

putpixel(x,y,WHITE);

}

getch();

closegraph();

}