PRACTICAL NO: 1

A. Study and enlist the basic functions used for graphics in C/C++ and python language.

Give an example for each of them

**A.Arc function in C**

Code:

#include<graphics.h>

#include<conio.h>

main() {

int gd=DETECT, gm;

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

arc(100,100,0,135,50);

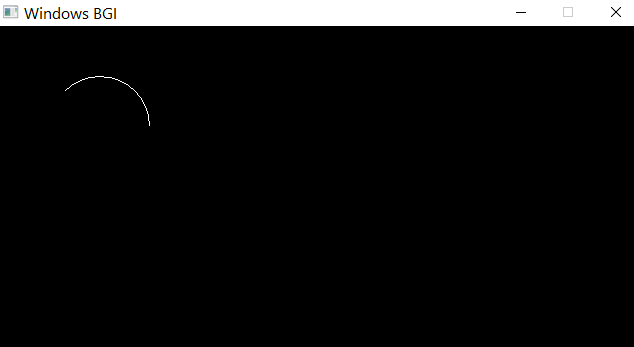
getch();

closegraph();

return 0;

}

Output:



**B.Drawpoly Function in C**

Code:

#include<graphics.h>

#include<conio.h>

main() {

int gd=DETECT,gm;

int points[]={320,150,420,300,250,300,320,150};

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

drawpoly(4,points);

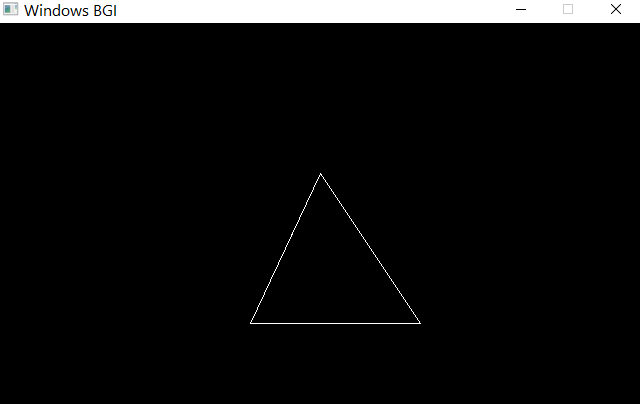
getch();

closegraph();

return 0;

}

Output:



**C.Closegraph function in C**

Code:

#include<graphics.h>

#include<conio.h>

main()

{

int gd=DETECT, gm;

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

outtext("Press any key to close the graph");

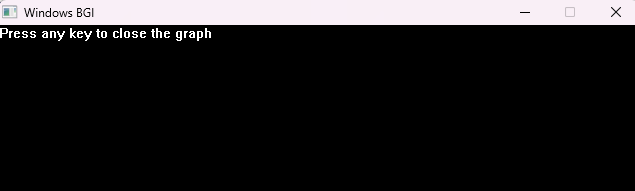
getch();

closegraph();

return 0;

}

OUTPUT:



**D.Circle function in C**

Code:

include<graphics.h>

#include<conio.h>

main()

{

int gd=DETECT, gm;

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

#include<graphics.h>

#include<conio.h>

main() {

int gd=DETECT, gm;

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

arc(100,100,0,135,50);

getch();

closegraph();

return 0;

}

circle(100,100,8);

getch();

closegraph();

return 0;

}

OUTPUT:



**E.Cleardevice function in C**

Code:

/c.Closegraph function in C

#include<graphics.h>

#include<conio.h>

main()

{

int gd=DETECT, gm;

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

outtext("Press any key to close the graph");

getch();

cleardevice();

outtext("Press any key to exit");

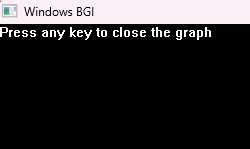
getch();

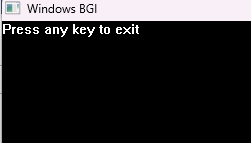
closegraph();

return 0;

}

OUTPUT:





**F.floodfill function in C**

Code:

#include<graphics.h>

#include<conio.h>

main()

{

int gd=DETECT, gm;

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

setcolor(RED);

circle(100,100,50);

floodfill(100,100,RED);

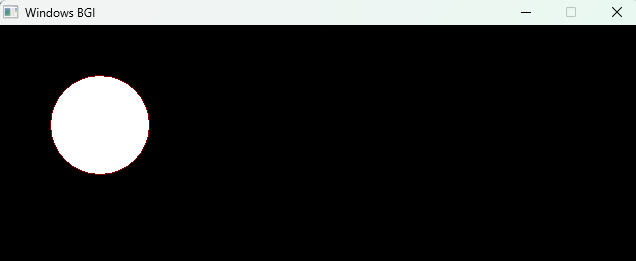
getch();

closegraph();

return 0;

}

OUTPUT:



**G.Getpixel function in C**

Code:

#include<graphics.h>

#include<conio.h>

main()

{

int gd=DETECT, gm;

char array[50];

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

int color=getpixel(0,0);

sprintf(array,"color of the pixel at (0,0)=%d",color);

outtext(array);

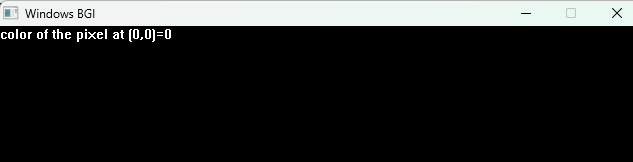
getch();

closegraph();

return 0;

}

OUTPUT:



**H.Set Color Function In C**

Code:

#include<graphics.h>

#include<conio.h>

main()

{

int gd=DETECT,gm;

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

outtext("Name:Birendar Chaurasiya Roll No:51\n");

circle(100,100,50);

setcolor(RED);

circle(200,200,50);

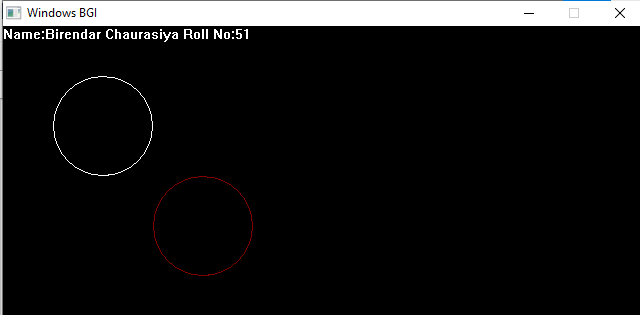
getch();

closegraph();

return 0;

}

OUTPUT:



PRACTICAL NO: 2

Aim : Divide your screen into four region,draw,circle,rectangle,ellipse and half ellipse in each region with appropriate message.

Code:

#include<graphics.h>

#include<conio.h>

main(){

int gd=DETECT,gm;

int poly[12]={350,450,350,410,430,400,350,350,300,430,350,450};

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

outtext("Name:Birendar Chaurasiya Roll No:51\n");

circle(100,100,50);

outtextxy(75,170,"Circle");

rectangle(200,50,350,150);

outtextxy(240,170,"Rectangle");

ellipse(500,100,0,360,100,50);

outtextxy(480,170,"Ellipse");

line(100,250,540,250);

outtextxy(300,260,"Line");

sector(150,400,30,300,100,50);

drawpoly(6,poly);

outtextxy(340,460,"Polygon");

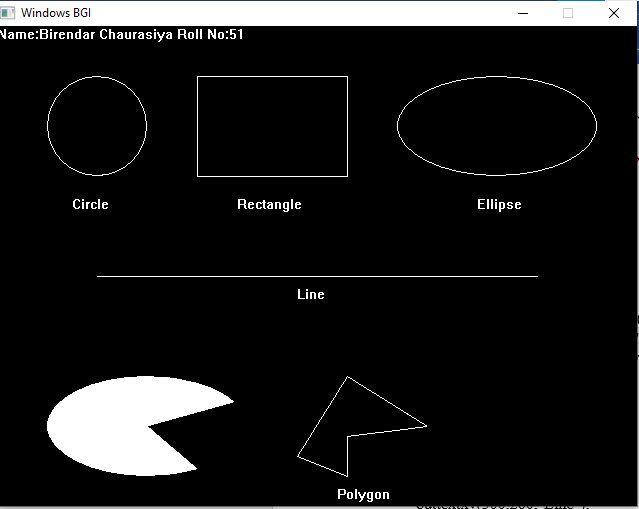
getch();

closegraph();

return 0;

}

OUTPUT:



B.Draw A Simple Hurt

Code:

#include<graphics.h>

#include<conio.h>

main(){

int gd= DETECT,gm;

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

outtextxy(250,0,"Name ishaque ROLLno 87");

setcolor(WHITE);

rectangle(150,180,250,300);

rectangle(250,180,420,300);

rectangle(180,250,220,300);

line(200,100,150,180);

line(200,100,250,180);

line(200,100,370,100);

line(370,100,420,180);

setfillstyle(SOLID\_FILL,BROWN);

floodfill(152,185,WHITE);

floodfill(252,182,WHITE);

setfillstyle(SLASH\_FILL,BLUE);

floodfill(182,252,WHITE);

setfillstyle(HATCH\_FILL,GREEN);

floodfill(200,105,WHITE);

floodfill(210,110,WHITE);

getch();

closegraph;

return 0;

}

Output:



PRACTICAL 3

1. Aim: Draw the basic circle in the center

Code:

#include <graphics.h>

#include<conio.h>

main(){

int gd=DETECT,gm;

int x,y,radius=80;

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

x=getmaxx()/2;

y=getmaxy()/2;

outtextxy(x-100,50,"CIRCLE USING GRAPHICS IN C");

outtext("Birendar chaurasiya/51");

circle(x,y,radius);

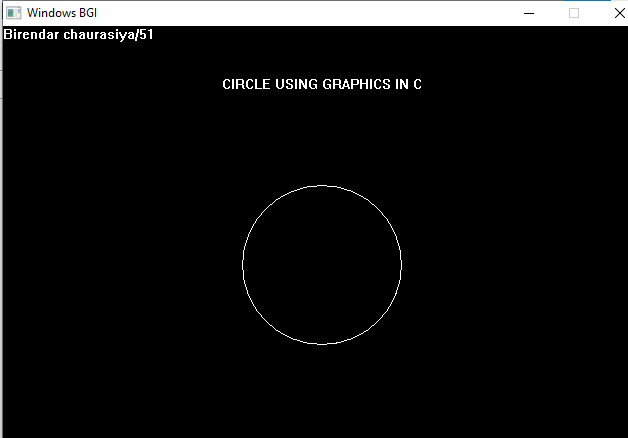
getch();

closegraph();

return 0;

}

Output:



B.

Code:

#include <graphics.h>

#include<conio.h>

main(){

int gd=DETECT,gm;

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

rectangle(150,50,400,150);

outtext("Birendar chaurasiya/51");

bar(150,200,400,350);

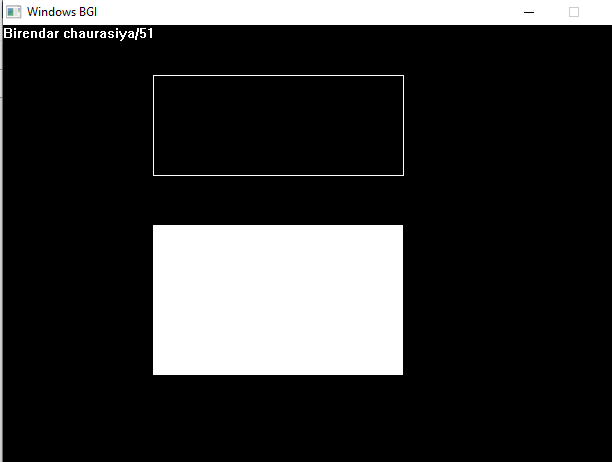
getch();

closegraph();

return 0;

}

Output:



C.

Code:

#include <graphics.h>

#include<conio.h>

main(){

int gd=DETECT,gm;

int x1=100,y1=250;

int x2=300,y2=250;

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

outtext("Birendar chaurasiya/51");

line(x1,y1,x2,y2);

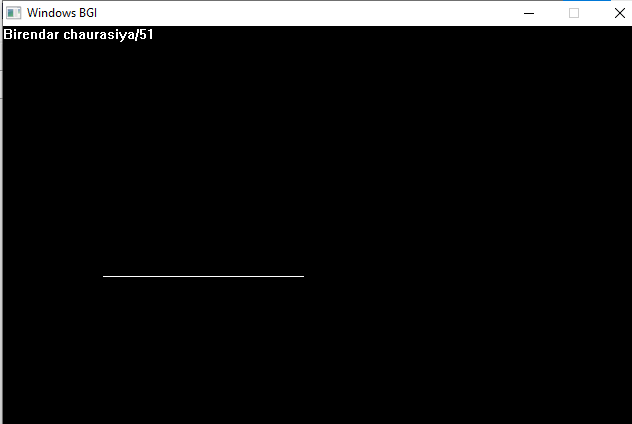
getch();

closegraph();

return 0;

}

Output:



D.

Code:

#include <graphics.h>

#include<conio.h>

main(){

int gd=DETECT,gm;

int x,y;

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

x=getmaxx()/2;

y=getmaxy()/2;

outtextxy(x-100,50,"ELLIPSE USING GRAPHICS IN C");

outtext("Birendar chaurasiya/51");

ellipse(x,y,0,360,120,60);

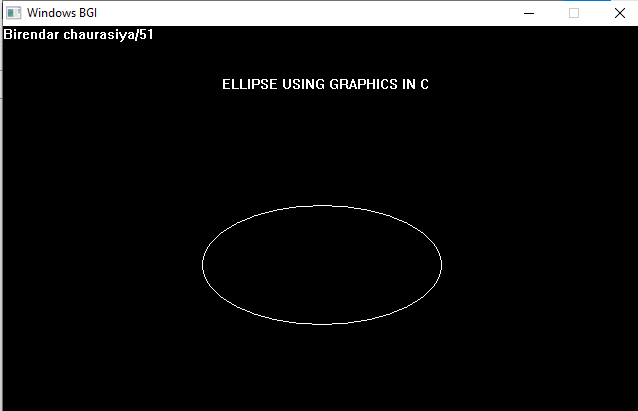
getch();

closegraph();

return 0;

}

Output:



E.

Code:

#include<graphics.h>

#include<math.h>

#include<conio.h>

#include<stdio.h>

int main(){

int rc,rb,xc,yc,i;

float x,y;

int gd=DETECT,gm;

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

printf("Enter the radius of the outer circle \n: ");

scanf(" %d",&rc);

printf("enter the radius of the inner circle \n: ");

scanf("%d",&rb);

printf("enter the center of the circle : \n: ");

scanf("%d",&xc);

yc=xc;

for( i=1;i<=360;i++){

x=xc+(rb\*(cos(i)));

y=yc+(rb\*(sin(i)));

putpixel(x,y,7);

}

for( i=1;i<=360;i++){

x=xc+(rc\*(cos(i)));

y=yc+(rc\*(sin(i)));

putpixel(x,y,7);

}

outtext("Birendar Chaurasiya 51");

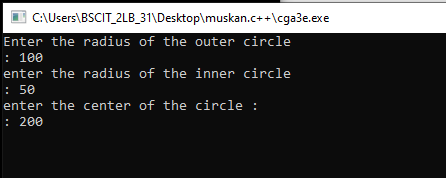
getch();

closegraph();

return 0;

}

Output:





PRACTICAL 4

1. DDA A line algorithm

Code:

#include <graphics.h>

#include<conio.h>

#include<math.h>

#include<iostream>

#include<dos.h>

using namespace std;

int main(){

int gd=DETECT,gm,i;

using std::cout;

float x,y,x1,y1,x2,y2,dx,dy,delx,dely,length;

cout<<endl<<"Birendar chaurasiya/51";

cout<<endl<<"Enter the co-ordinates of starting point :";

cin>>x1>>y1>>x2>>y2;

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

dx=x2-x1;

dy=y2-y1;

if(dx>=dy){

length=dx;

}

else

{

length=dy;

}

delx=dx/length;

dely=dy/length;

x=x1+0.5;

y=y1+0.5;

i=1;

while(i<=length)

{

putpixel(x,y,5);

x=x+delx;

y=y+dely;

i=i+1;

delay(50);

}

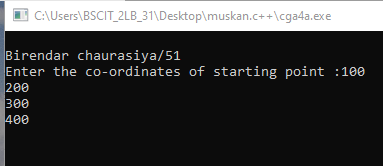
getch();

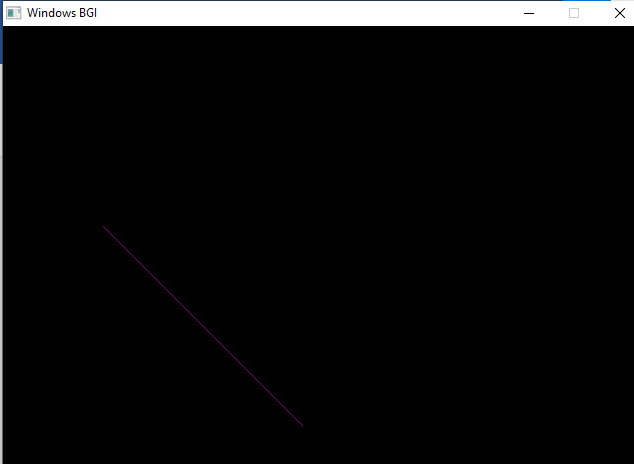
closegraph();

return 0;

}

Output:





1. Bresenham’s Line Algorithm

Code:

#include<graphics.h>

#include<conio.h>

#include<math.h>

#include<dos.h>

#include<iostream>

using namespace std;

main() {

int gd=DETECT,gm,i;

using std::cout;

float x,y,x1,y1,x2,y2,dx,dy,e;

cout<<endl<<"BIRENDAR CHAURASIYA 51\n";

cout<<endl<<"Enter the co-ordinate of starting point:";

cin>>x1>>y1>>x2>>y2;

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

dx=abs(x2-x1);

dy=abs(y2-y1);

x=x1;

y=y1;

e=2\*(dy-dx);

i=1;

do

{

putpixel(x,y,GREEN);

while(e>=0){

y=y+1;

e=e-2\*dx;

}

x=x+1;

e=e+2\*dy;

i=i+1;

delay(50);

}

while(i<=dx);

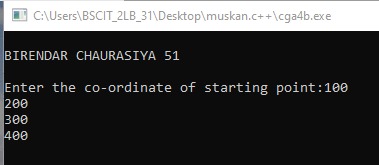
getch();

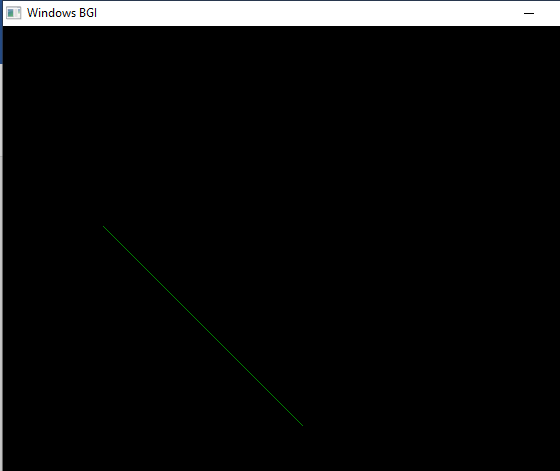
closegraph();

return 0;

}

Output:





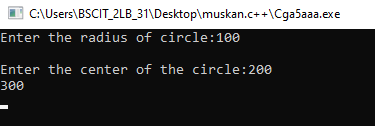
PRACTICAL NO 5

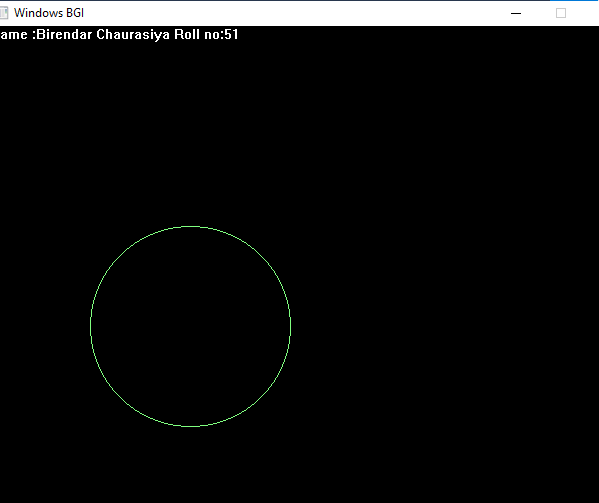
A.Midpoint circle alogrithm

Code:

#include<graphics.h>  
#include<conio.h>  
#include<math.h>  
#include<dos.h>  
#include<iostream>  
using namespace std;  
int main(){  
int gd=DETECT,gm,i;  
using std::cout;  
float r,x,y,xc,yc,p,d;  
initgraph(&gd,&gm,"C:\\TC\\BGI");  
outtext("Name :Birendar Chaurasiya Roll no:51");  
cout<<"Enter the radius of circle:";  
cin>>r;  
cout<<endl<<"Enter the center of the circle:";  
cin>>xc>>yc;  
x=0;  
y=r;  
d=1.25-r;  
do{  
putpixel(x+xc,y+yc,10);  
putpixel(y+xc,x+yc,10);  
putpixel(-x+xc,y+yc,10);  
putpixel(-y+xc,x+yc,10);  
putpixel(-x+xc,-y+yc,10);  
putpixel(-y+xc,-x+yc,10);  
putpixel(x+xc,-y+yc,10);  
putpixel(y+xc,-x+yc,10);  
  
if(d<0)  
{  
x=x+1;  
y=y;  
d=d+2\*x+1;  
}else  
{  
x=x+1;  
y=y-1;  
d=d+2\*(x-y)+1;  
}  
delay(50);  
}  
while(x<y);  
getch();  
closegraph();  
return 0;  
}

Output:





Practical No .6

Aim: Write a program to implement 2D scaling

Code:

#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<conio.h>

#include<math.h>

int main()

{

int gd=DETECT,gm;

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

int sx,sy,xt,yt,r;

float t;

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

printf(" \t Program for basic transactions");

printf(" \n \t Enter the points of triangle:");

setcolor(1);

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

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

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

printf("\n Enter your choice:");

scanf("%d",&c);

switch(c)

{

case 1: cleardevice();

printf(" n Enter the translation factor:");

scanf("%d%d",&xt,&yt);

nx1=x1+xt;

ny1=y1+yt;

nx2=x2+xt;

ny2=y2+yt;

nx3=x3+xt;

ny3=y3+yt;

line(nx1,ny1,nx2,ny2);

line(nx2,ny2,nx3,ny3);

line(nx3,ny3,nx1,ny1);

getch();

break;

case 2: cleardevice();

printf(" \ n Enter the angle of rotation:");

scanf("%d",&r);

t=3.14\*r/180;

nx1=abs(x1\*cos(t)-y1\*sin(t));

ny1=abs(x1\*sin(t)+y1\*cos(t));

nx2=abs(x2\*cos(t)-y2\*sin(t));

ny2=abs(x2\*sin(t)+y2\*cos(t));

nx3=abs(x3\*cos(t)-y3\*sin(t));

ny3=abs(x3\*sin(t)+y3\*cos(t));

line(nx1,ny1,nx2,ny2);

line(nx2,ny2,nx3,ny3);

line(nx3,ny3,nx1,ny1);

getch();

break;

case 3: cleardevice();

printf(" \n Enter the scalling factor:");

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

nx1=x1\*sx;

ny1=y2\*sy;

nx2=x2\*sx;

ny2=y2\*sy;

nx3=x3\*sx;

ny3=y3\*sy;

line(nx1,ny1,nx2,ny2);

line(nx2,ny2,nx3,ny3);

line(nx3,ny3,nx1,ny1);

getch();

case 4:

break;

default:

printf("Enter the correct choice:");

}

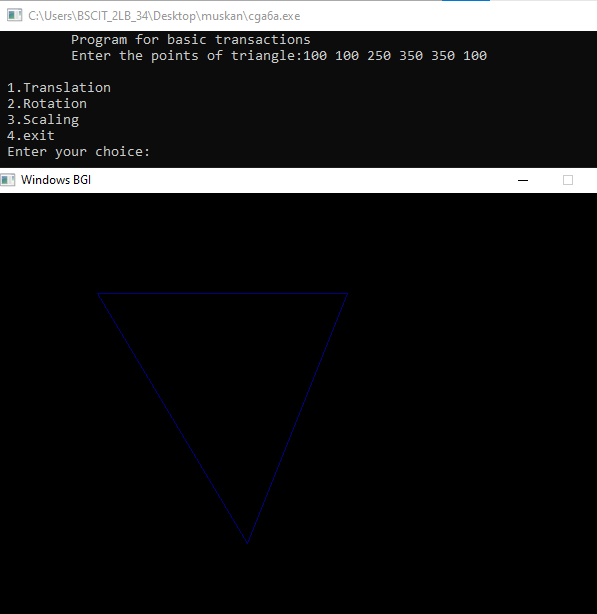
closegraph();

return 0;

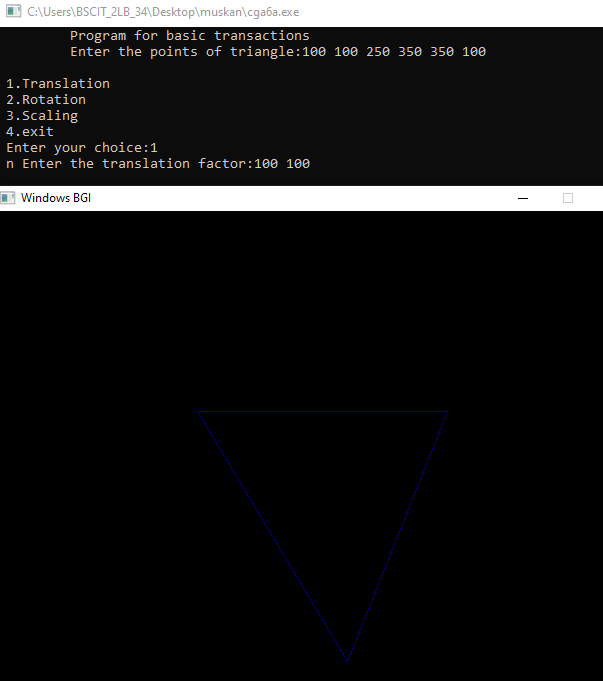
}

Output:

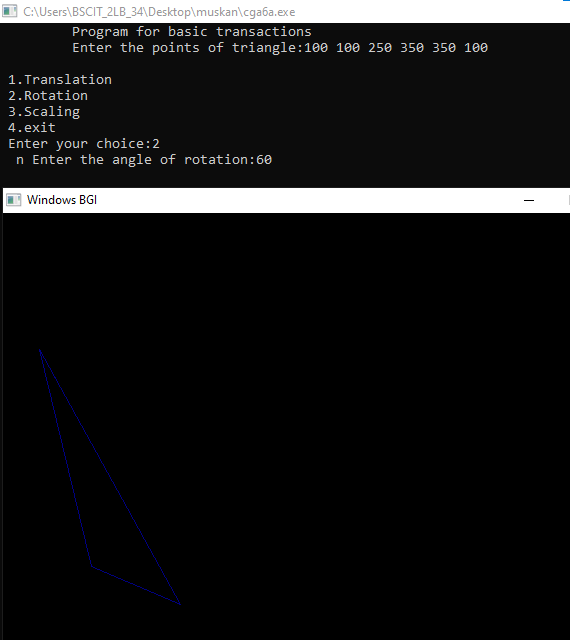
Before: Translation



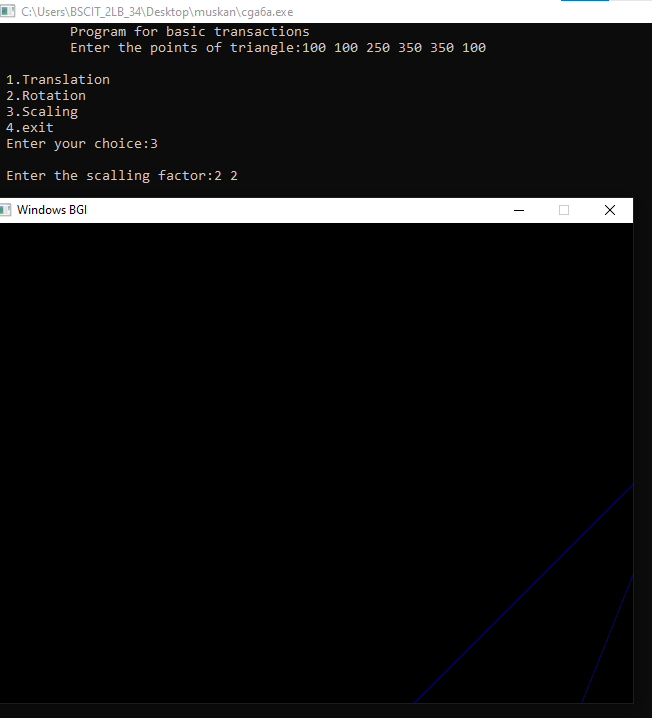
After Translation



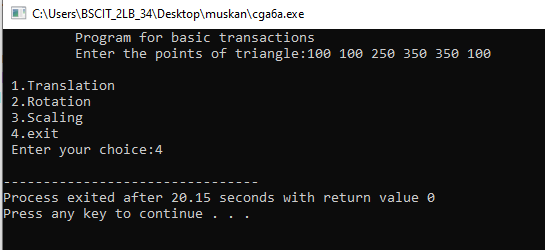
2.Rotation



3. Scaling



4.Exit



PRACTICAL NO 7

AIM:Perform 2D rotation on a given object

CODE:

#include<stdio.h>

#include<math.h>

#include<conio.h>

//Using macros to convert degree to radian

//and call sin() and cos() as these functions

//take input in radians

#define SIN(x) sin(x\*3.141592653589/180)

#define COS(x) cos(x\*3.141592653589/180)

//To rotate an object

void rotate(float a[][2],int n,int x\_pivot,int y\_pivot,int angle)

{

int i=0;

while(i<n)

{

//Shifting the pivot point to the origin

//and the given points accordingly

int x\_shifted=a[i][0]-x\_pivot;

int y\_shifted=a[i][1]-y\_pivot;

//Calculating the rotated point co-ordinates

//and shifting it back

a[i][0]=x\_pivot + (x\_shifted\*COS(angle)-y\_shifted\*SIN(angle));

a[i][1]=y\_pivot + (x\_shifted\*SIN(angle)+y\_shifted\*COS(angle));

printf("(%f,%f)",a[i][0],a[i][1]);

i++;

}

}

//Driver Code

int main()

{

//1st Example

//The following figure is to be

//rotated about (0,0) by 90 degrees

int size1=4; //No. of vertices

//Vertex co-ordinates must be in order

float points\_list1[][2]={{100,100},{150,200},{200,200},{200,150}};

rotate(points\_list1,size1,0,0,90);

//2nd Example

//The following figure is to be

//rotated about (50,-50) by -45 degrees

/\*int size2=3; //No. of vertices

float points\_list2[][2]={{100,100},{100,200},{200,200}};

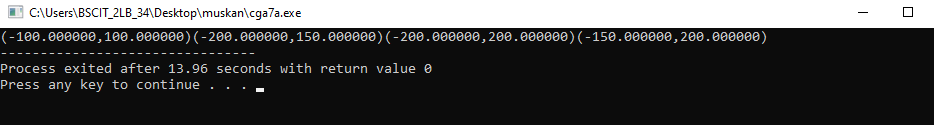
rotate(points\_list2,size2,50,-50,-45);\*/

getch();

return 0;

}

OUTPUT:



Practical No.8

Aim:Sutherland Cohen Subdivision Line Clipping

Code:

#include<iostream>

#include<stdlib.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

using namespace std;

typedef struct coordinate

{

int x,y;

char code[4];

}PT;

void drawwindow();

void drawline(PT p1,PT p2);

PT setcode(PT p);

int visibility(PT p1,PT p2);

PT resetendpt(PT p1,PT p2);

int main()

{

using std::cout;

int gd=DETECT,v,gm;

PT p1,p2,p3,p4,ptemp;

cout<<"\nEnter x1 and y1\n";

cin>>p1.x>>p1.y;

cout<<"\nEnter x2 and y2\n";

cin>>p2.x>>p2.y;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

drawwindow();

delay(500);

drawline(p1,p2);

delay(500);

cleardevice();

delay(500);

p1=setcode(p1);

p2=setcode(p2);

v=visibility(p1,p2);

delay(500);

switch(v)

{

case 0: drawwindow();

delay(500);

drawline(p1,p2);

break;

case 1: drawwindow();

delay(500);

break;

case 2: p3=resetendpt(p1,p2);

p4=resetendpt(p2,p1);

drawwindow();

delay(500);

drawline(p3,p4);

break;

}

delay(5000);

closegraph();

return 0;

}

void drawwindow()

{

line(150,100,450,100);

line(450,100,450,350);

line(450,350,150,350);

line(150,350,150,100);

}

void drawline(PT p1,PT p2)

{

line(p1.x,p1.y,p2.x,p2.y);

}

PT setcode(PT p) //for setting the 4 bit code

{

PT ptemp;

if(p.y<100)

ptemp.code[0]='1'; //Top

else

ptemp.code[0]='0';

if(p.y>350)

ptemp.code[1]='1'; //Bottom

else

ptemp.code[1]='0';

if(p.x>450)

ptemp.code[2]='1'; //Right

else

ptemp.code[2]='0';

if(p.x<150)

ptemp.code[3]='1'; //Left

else

ptemp.code[3]='0';

ptemp.x=p.x;

ptemp.y=p.y;

return(ptemp);

}

int visibility(PT p1,PT p2)

{

int i,flag=0;

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

{

if((p1.code[i]!='0') || (p2.code[i]!='0'))

flag=1;

}

if(flag==0)

return(0);

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

{

if((p1.code[i]==p2.code[i]) && (p1.code[i]=='1'))

flag='0';

}

if(flag==0)

return(1);

return(2);

}

PT resetendpt(PT p1,PT p2)

{

PT temp;

int x,y,i;

float m,k;

if(p1.code[3]=='1')

x=150;

if(p1.code[2]=='1')

x=450;

if((p1.code[3]=='1') || (p1.code[2]=='1'))

{

m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(p1.y+(m\*(x-p1.x)));

temp.y=k;

temp.x=x;

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

temp.code[i]=p1.code[i];

if(temp.y<=350 && temp.y>=100)

return (temp);

}

if(p1.code[0]=='1')

y=100;

if(p1.code[1]=='1')

y=350;

if((p1.code[0]=='1') || (p1.code[1]=='1'))

{

m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(float)p1.x+(float)(y-p1.y)/m;

temp.x=k;

temp.y=y;

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

temp.code[i]=p1.code[i];

return(temp);

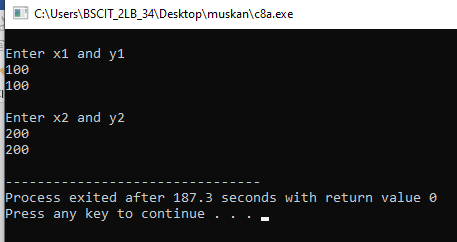
}

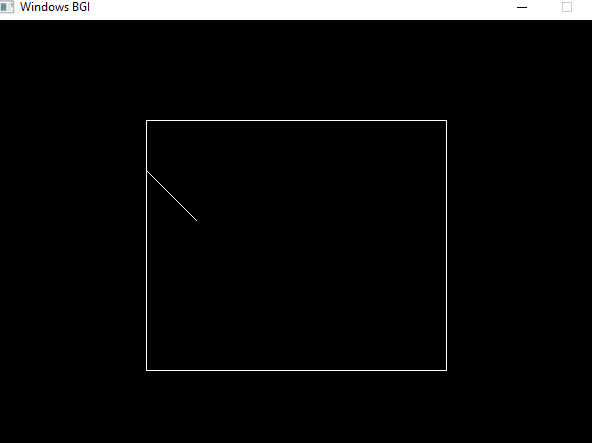
else

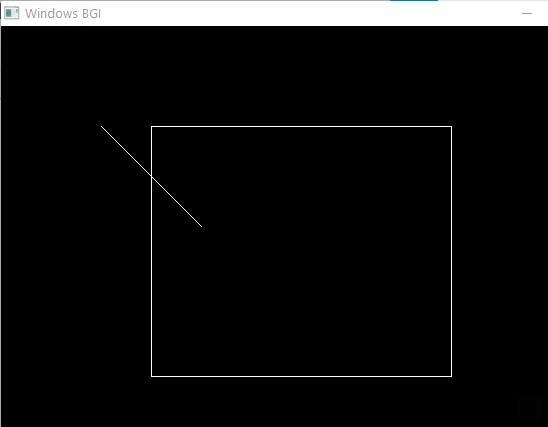
return(p1);

}

Output:







PRACTICAL NO 9

Aim: FloodFill Algorithm

Code:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

void floodFill(int, int, int, int);

int midx=319, midy=239;

int main()

{

int gdriver=DETECT, gmode, x,y,r;

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

outtext("Birendar Chaurasiya 51");

printf("Enter the Center of circle (x,y) : ");

scanf("%d %d",&x,&y);

printf("Enter the Radius of circle r : ");

scanf("%d",&r);

circle(midx+x,midy-y,r);

getch();

floodFill(midx+x,midy-y,14,0);

getch();

closegraph();

return 0;

}

void floodFill(int x, int y, int fill, int old)

{

if(getpixel(x,y) == old)

{

putpixel(x,y,fill);

delay(5);

floodFill(x+1,y,fill,old);

floodFill(x-1,y,fill,old);

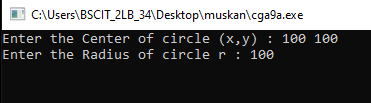
floodFill(x,y+1,fill,old);

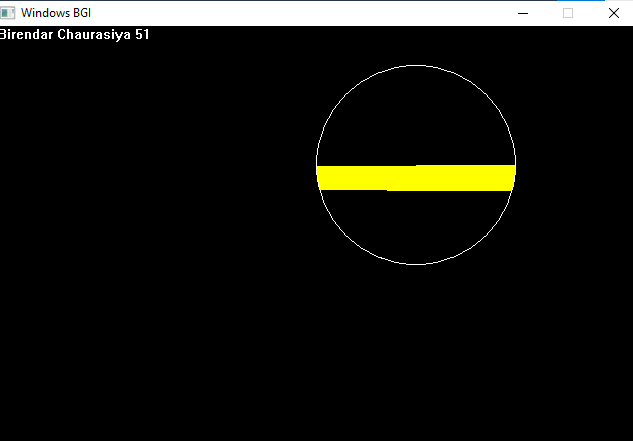
floodFill(x,y-1,fill,old);

}

}

Output:





PRACTICAL NO 10

Aim:Draw the moving car on the screen

Code:

#include<graphics.h>

#include<conio.h>

#include<math.h>

#include<iostream>

#include<dos.h>

int main() {

int gd=DETECT, gm;

int i,maxx,midy;

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

outtext("Birendar Chaurasiya 51");

maxx=getmaxx();

midy=getmaxy()/2;

for(i=0; i < maxx-150; i=i+5)

{

cleardevice();

setcolor(WHITE);

line(0,midy+37,maxx,midy+37);

setcolor(YELLOW);

setfillstyle(SOLID\_FILL,RED);

line(i ,midy +23, i, midy);

line(i,midy,40 + i, midy-20);

line(40 + i,midy-20,80 + i, midy-20);

line(80 + i,midy-20,100 + i, midy);

line(100 + i,midy,120 + i, midy);

line(120 + i,midy,120 + i, midy+23);

line(0 + i,midy+ 23,18 + i, midy+23);

arc(30 + i,midy+23,0,180,12);

line(42 + i,midy+23,78 + i, midy+23);

arc(90 + i,midy+23,0,180,12);

line(102 + i,midy+23,120 + i, midy+23);

line(28 + i,midy,43 + i, midy-15);

line(43 + i,midy-15,57 + i, midy-15);

line(57 + i,midy-15,57 + i, midy);

line(57 + i,midy,28 + i, midy);

line(62 + i,midy-15,77 + i, midy-15);

line(77 + i,midy-15,92 + i, midy);

line(92 + i,midy,62 + i, midy);

line(62+i,midy,62+i,midy-15);

floodfill(5+i,midy+22,YELLOW);

setcolor(BLUE);

setfillstyle(SOLID\_FILL,DARKGRAY);

circle(30+i,midy+25,9);

circle(90+i,midy+25,9);

floodfill(30+i,midy+25,BLUE);

floodfill(90+i,midy+25,BLUE);

delay(100);

getch();

}

closegraph();

return 0;

}

Output:

