Computer Graphics Lab 2

Experiment No: 1.

Experiment Name: Scan converting a circle with midpoint circle algorithm.

Code:

#include <iostream>

#include <graphics.h>

void drawCircle(int xc, int yc, int radius)

{

int x = 0;

int y = radius;

int d = 1 - radius;

while (x <= y)

{

// Plot the symmetric pixels in each octant

putpixel(xc + x, yc + y, WHITE); // Octant 1

putpixel(xc - x, yc + y, WHITE); // Octant 2

putpixel(xc + x, yc - y, WHITE); // Octant 8

putpixel(xc - x, yc - y, WHITE); // Octant 7

putpixel(xc + y, yc + x, WHITE); // Octant 3

putpixel(xc - y, yc + x, WHITE); // Octant 4

putpixel(xc + y, yc - x, WHITE); // Octant 6

putpixel(xc - y, yc - x, WHITE); // Octant 5

// Update the decision parameter and coordinates

if (d <= 0)

{

d += 2 \* x + 1;

x++;

}

else

{

d += 2 \* (x - y) + 1;

x++;

y--;

}

}

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, ""); // Initialize graphics mode

int centerX = 200; // x-coordinate of the center

int centerY = 200; // y-coordinate of the center

int radius = 100; // radius of the circle

drawCircle(centerX, centerY, radius);

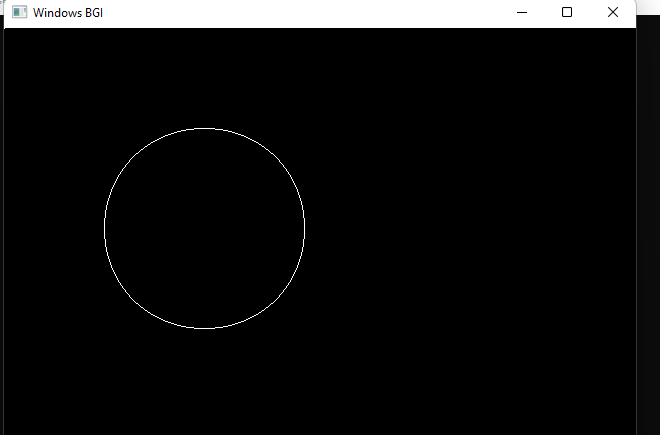
getch(); // Wait for a key press

closegraph(); // Close graphics mode

return 0;

}

**Output:**



Experiment No: 2.

Experiment Name: Scan converting an ellipse with midpoint algorithm.

Code:

# include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void disp();

float x,y;

int xc,yc;

int main()

{

int gd=DETECT,gm,a,b;

float p1,p2;

//clrscr();

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

printf("\*\*\* Ellipse Generating Algorithm \*\*\*\n");

printf("Enter the value of Xc\t");

scanf("%d",&xc);

printf("Enter the value of yc\t");

scanf("%d",&yc);

printf("Enter X axis length\t");

scanf("%d",&a);

printf("Enter Y axis length\t");

scanf("%d",&b);

x=0;y=b;

disp();

p1=(b\*b)-(a\*a\*b)+(a\*a)/4;

while((2.0\*b\*b\*x)<=(2.0\*a\*a\*y))

{

x++;

if(p1<=0)

p1=p1+(2.0\*b\*b\*x)+(b\*b);

else

{

y--;

p1=p1+(2.0\*b\*b\*x)+(b\*b)-(2.0\*a\*a\*y);

}

disp();

x=-x;

disp();

x=-x;

delay(50);

}

x=a;

y=0;

disp();

p2=(a\*a)+2.0\*(b\*b\*a)+(b\*b)/4;

while((2.0\*b\*b\*x)>(2.0\*a\*a\*y))

{

y++;

if(p2>0)

p2=p2+(a\*a)-(2.0\*a\*a\*y);

else

{

x--;

p2=p2+(2.0\*b\*b\*x)-(2.0\*a\*a\*y)+(a\*a);

}

disp();

y=-y;

disp();

y=-y;

delay(50);

}

getch();

closegraph();

}

void disp()

{

putpixel(xc+x,yc+y,7);

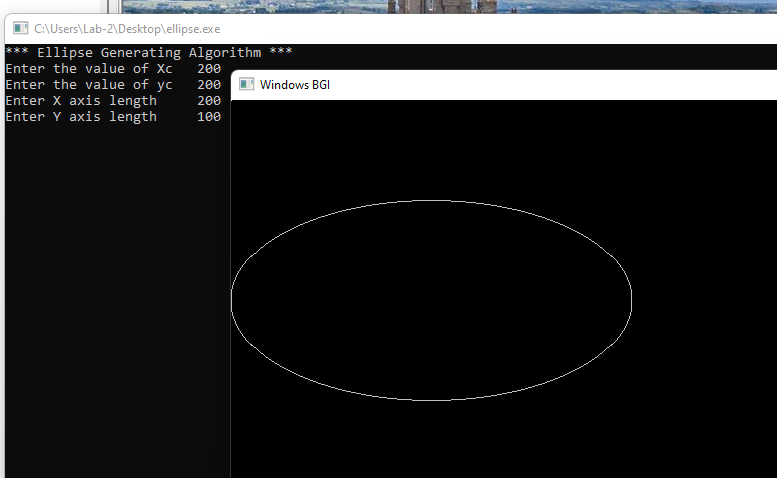
putpixel(xc-x,yc+y,7);

putpixel(xc+x,yc-y,7);

putpixel(xc+x,yc-y,7);

}

**Output:**



---\*---