Assignment - 2

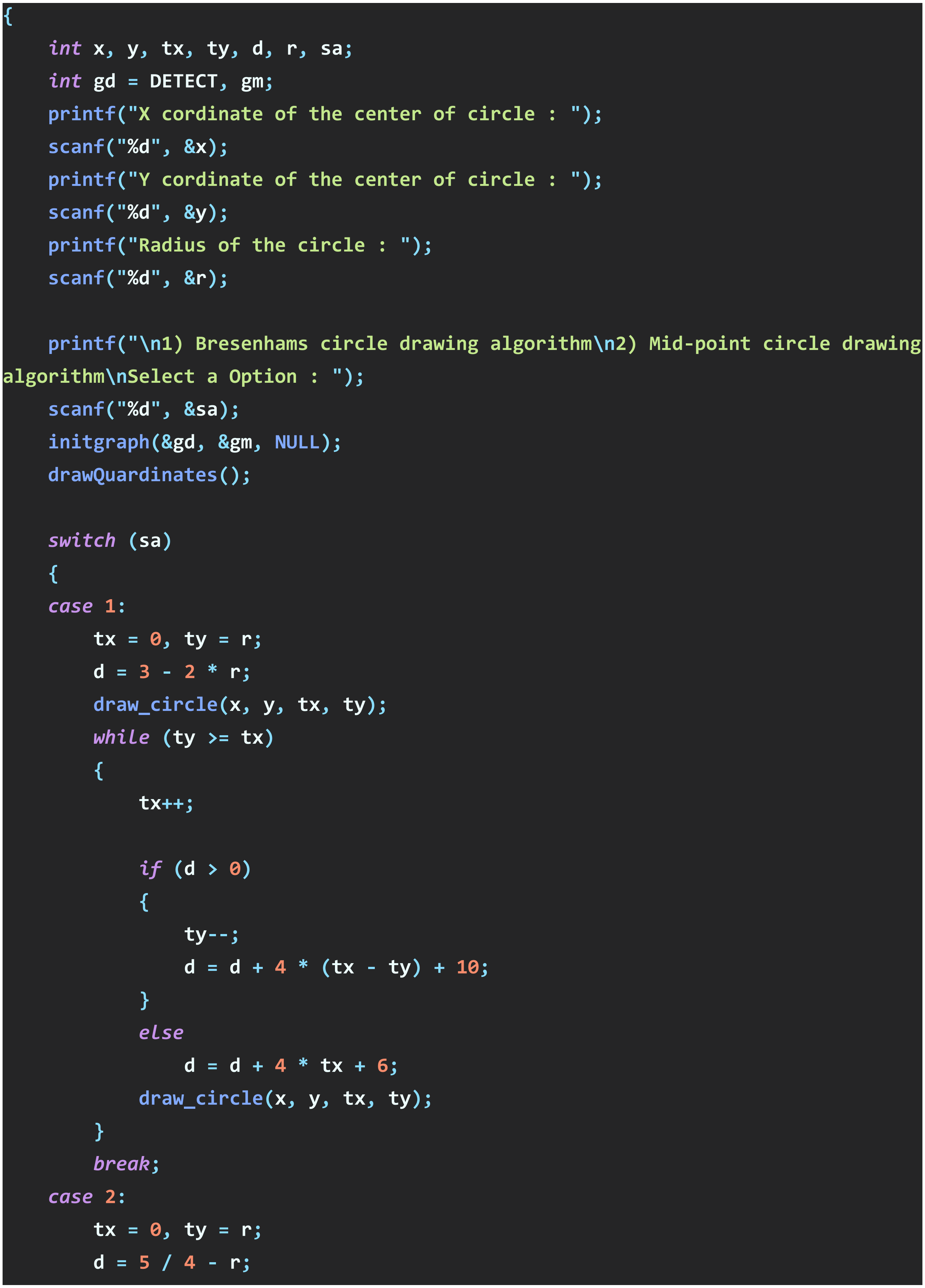
Name : Emmadi Sumith Kumar
Branch : CSE - B1
Subject : Computer Graphics
Roll No : UI20CS21

1. Write a program to implement Circle Drawing Algorithm using switch case
   1. Bresenham's
   2. Mid Point

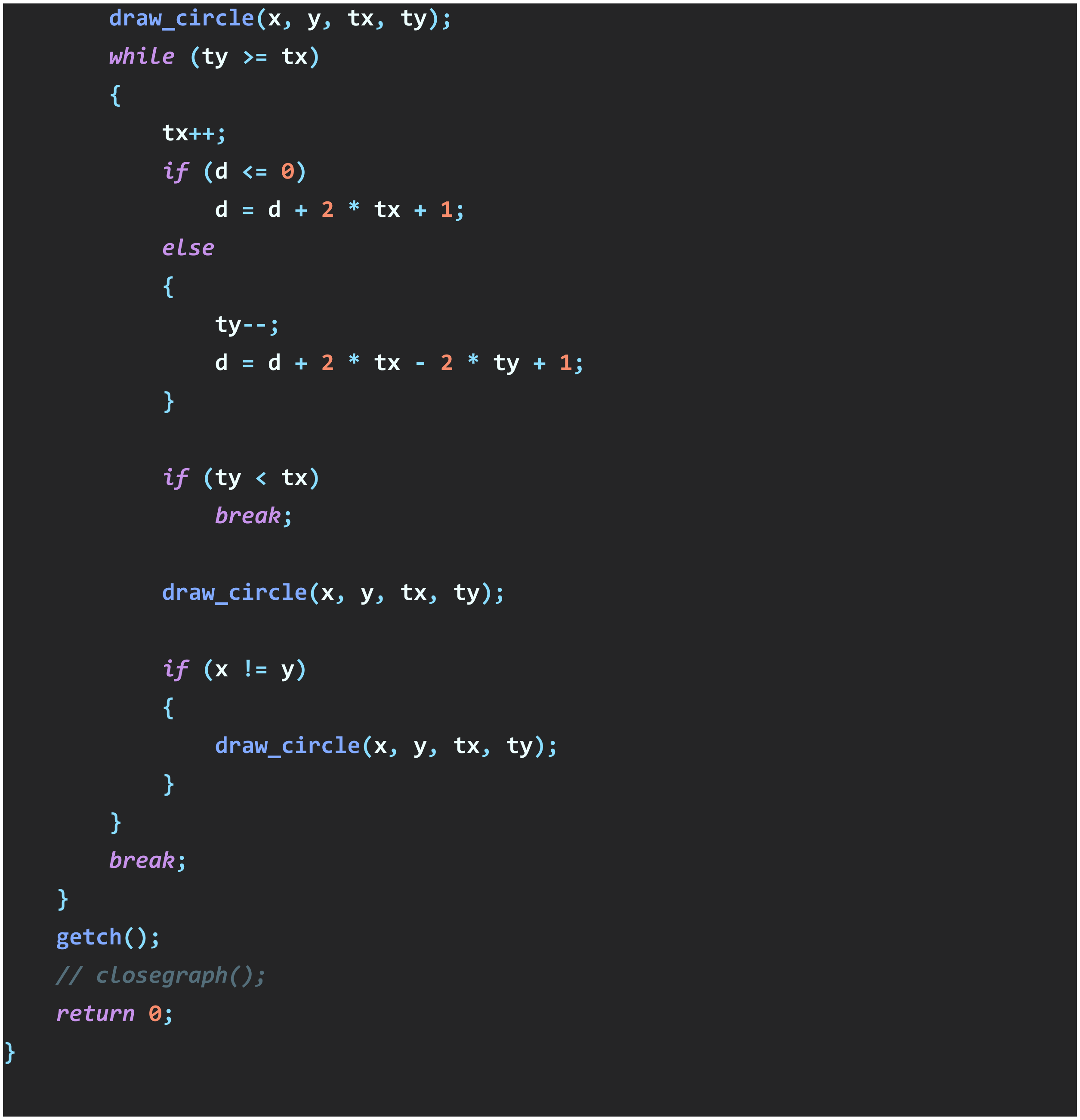
2. Write a program to implement Mid Point Ellipse Drawing Algorithm

1. Write a program to implement Circle Drawing Algorithm using switch case
   1. Bresenham's
   2. Mid Point

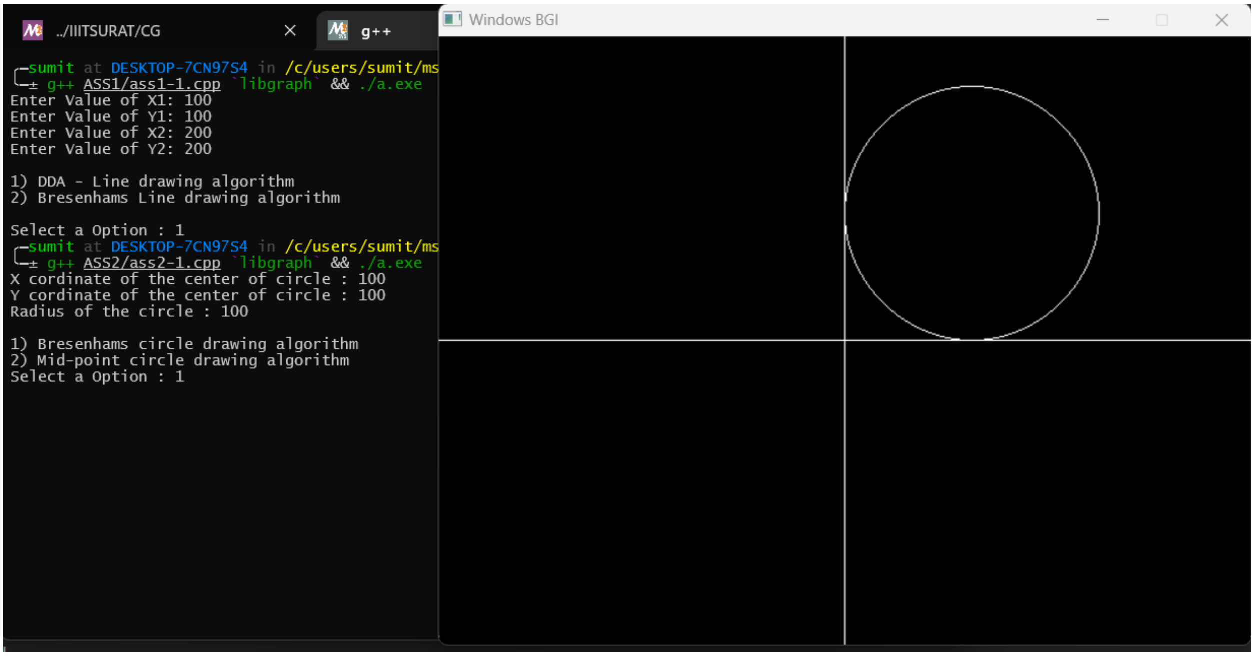
```
#include <stdio.h>
#include <dos.h>
#include <graphics.h>

void drawQuardinates()
{
    line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
    line(getmaxx() / 2, 0, getmaxx() / 2, getmaxy());
}

void draw_circle(int x, int y, int tx, int ty)
{
    putpixel(getmaxx() / 2 + x + tx, getmaxy() / 2 - (y + ty), WHITE);
    putpixel(getmaxx() / 2 + x - tx, getmaxy() / 2 - (y + ty), WHITE);
    putpixel(getmaxx() / 2 + x + tx, getmaxy() / 2 - (y - ty), WHITE);
    putpixel(getmaxx() / 2 + x - tx, getmaxy() / 2 - (y - ty), WHITE);
    putpixel(getmaxx() / 2 + x + ty, getmaxy() / 2 - (y + tx), WHITE);
    putpixel(getmaxx() / 2 + x - ty, getmaxy() / 2 - (y + tx), WHITE);
    putpixel(getmaxx() / 2 + x + ty, getmaxy() / 2 - (y - tx), WHITE);
    putpixel(getmaxx() / 2 + x - ty, getmaxy() / 2 - (y - tx), WHITE);
}

int main()
```



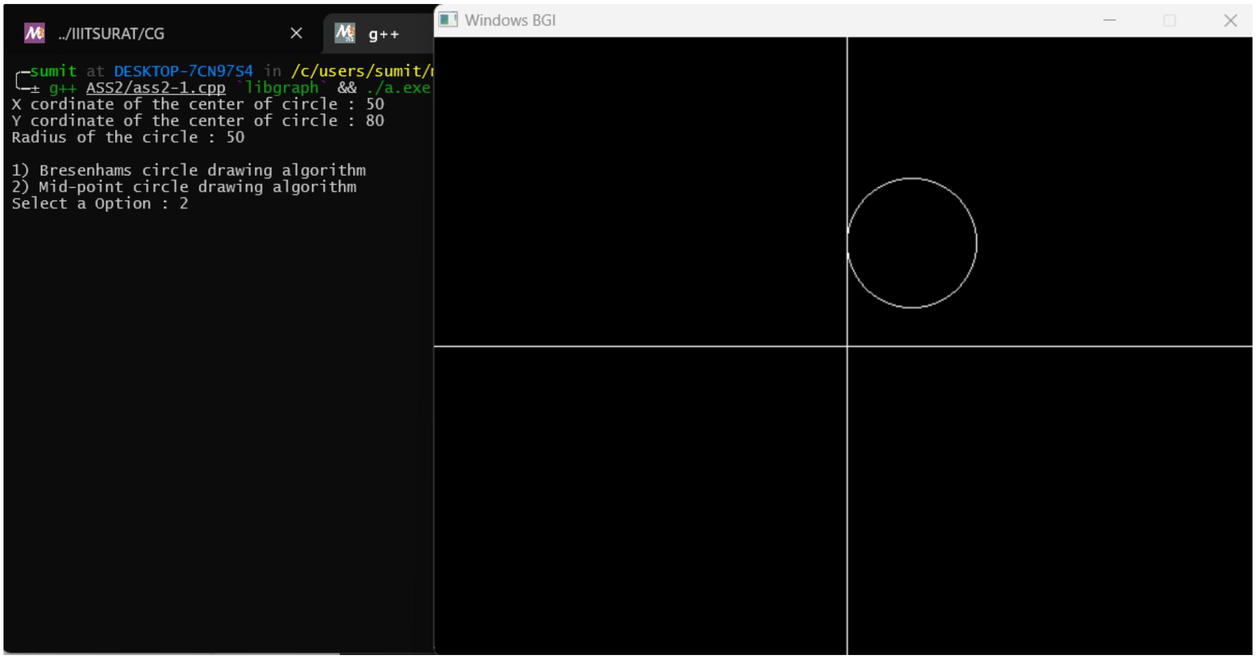
```
{
    int x, y, tx, ty, d, r, sa;
    int gd = DETECT, gm;
    printf("X cordinate of the center of circle : ");
    scanf("%d", &x);
    printf("Y cordinate of the center of circle : ");
    scanf("%d", &y);
    printf("Radius of the circle : ");
    scanf("%d", &r);

    printf("\n1) Bresenhams circle drawing algorithm\n2) Mid-point circle drawing
algorithm\nSelect a Option : ");
    scanf("%d", &sa);
    initgraph(&gd, &gm, NULL);
    drawQuardinates();

    switch (sa)
    {
    case 1:
        tx = 0, ty = r;
        d = 3 - 2 * r;
        draw_circle(x, y, tx, ty);
        while (ty >= tx)
        {
            tx++;

            if (d > 0)
            {
                ty--;
                d = d + 4 * (tx - ty) + 10;
            }
            else
                d = d + 4 * tx + 6;
            draw_circle(x, y, tx, ty);
        }
        break;
    case 2:
        tx = 0, ty = r;
        d = 5 / 4 - r;
```



```
        draw_circle(x, y, tx, ty);
        while (ty >= tx)
        {
            tx++;
            if (d <= 0)
                d = d + 2 * tx + 1;
            else
            {
                ty--;
                d = d + 2 * tx - 2 * ty + 1;
            }

            if (ty < tx)
                break;

            draw_circle(x, y, tx, ty);

            if (x != y)
            {
                draw_circle(x, y, tx, ty);
            }
        }
        break;
    }
    getch();
    // closegraph();
    return 0;
}
```



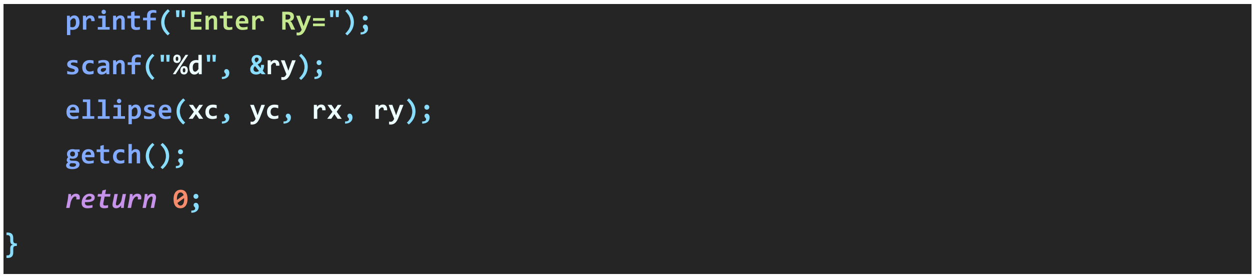
```
╭─sumit at DESKTOP-7CN97S4 in /c/users/sumit/ms
╰─± g++ ASS1/ass1-1.cpp `libgraph` && ./a.exe
Enter Value of X1: 100
Enter Value of Y1: 100
Enter Value of X2: 200
Enter Value of Y2: 200

1) DDA - Line drawing algorithm
2) Bresenhams Line drawing algorithm

Select a Option : 1
╭─sumit at DESKTOP-7CN97S4 in /c/users/sumit/ms
╰─± g++ ASS2/ass2-1.cpp `libgraph` && ./a.exe
X cordinate of the center of circle : 100
Y cordinate of the center of circle : 100
Radius of the circle : 100

1) Bresenhams circle drawing algorithm
2) Mid-point circle drawing algorithm
Select a Option : 1
```



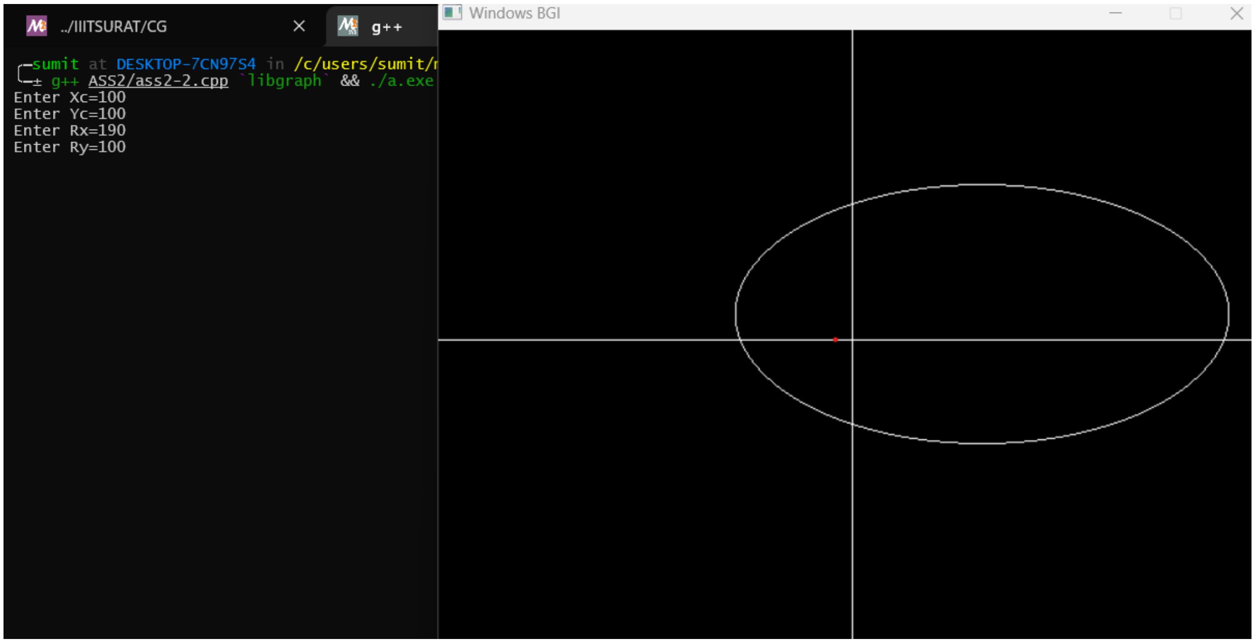
```
╭─sumit at DESKTOP-7CN97S4 in /c/users/sumit/r
╰─± g++ ASS2/ass2-1.cpp `libgraph` && ./a.exe
X cordinate of the center of circle : 50
Y cordinate of the center of circle : 80
Radius of the circle : 50

1) Bresenhams circle drawing algorithm
2) Mid-point circle drawing algorithm
Select a Option : 2
```

2. Write a program to implement Mid Point Ellipse Drawing Algorithm

```c
#include <stdio.h>
#include <conio.h>
#include <graphics.h>

void drawQuardinates()
{
    line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
    line(getmaxx() / 2, 0, getmaxx() / 2, getmaxy());
}

void ellipse(int xc, int yc, int rx, int ry)
{
    int gm = DETECT, gd;
    int x, y, p;

    initgraph(&gm, &gd, NULL);
    drawQuardinates();
    x = 0;
    y = ry;
    p = (ry * ry) - (rx * rx * ry) + ((rx * rx) / 4);
    while ((2 * x * ry * ry) < (2 * y * rx * rx))
    {
        putpixel(getmaxx() / 2 + xc + x, getmaxx() / 2 - (yc - y), WHITE);
        putpixel(getmaxx() / 2 + xc - x, getmaxx() / 2 - (yc + y), WHITE);
        putpixel(getmaxx() / 2 + xc + x, getmaxx() / 2 - (yc + y), WHITE);
        putpixel(getmaxx() / 2 + xc - x, getmaxx() / 2 - (yc - y), WHITE);

        if (p < 0)
        {
            x = x + 1;
            p = p + (2 * ry * ry * x) + (ry * ry);
        }
        else
        {
            x = x + 1;
            y = y - 1;
            p = p + (2 * ry * ry * x + ry * ry) - (2 * rx * rx * y);
        }
    }
    p = ((float)x + 0.5) * ((float)x + 0.5) * ry * ry + (y - 1) * (y - 1) * rx *
rx - rx * rx * ry * ry;

    while (y >= 0)
    {
        putpixel(getmaxx() / 2 + xc + x, getmaxx() / 2 - (yc - y), WHITE);
        putpixel(getmaxx() / 2 + xc - x, getmaxx() / 2 - (yc + y), WHITE);
        putpixel(getmaxx() / 2 + xc + x, getmaxx() / 2 - (yc + y), WHITE);
        putpixel(getmaxx() / 2 + xc - x, getmaxx() / 2 - (yc - y), WHITE);

        if (p > 0)
        {
            y = y - 1;
            p = p - (2 * rx * rx * y) + (rx * rx);
        }
        else
        {
            y = y - 1;
            x = x + 1;
            p = p + (2 * ry * ry * x) - (2 * rx * rx * y) - (rx * rx);
        }
    }
    getch();
    closegraph();
}

int main()
{
    int xc, yc, rx, ry;

    printf("Enter Xc=");
    scanf("%d", &xc);
    printf("Enter Yc=");
    scanf("%d", &yc);
    printf("Enter Rx=");
    scanf("%d", &rx);
```



```
    printf("Enter Ry=");
    scanf("%d", &ry);
    ellipse(xc, yc, rx, ry);
    getch();
    return 0;
}
```



```
╭─sumit at DESKTOP-7CN97S4 in /c/users/sumit/r
╰─± g++ ASS2/ass2-2.cpp `libgraph` && ./a.exe
Enter Xc=100
Enter Yc=100
Enter Rx=190
Enter Ry=100
```