Lab No: 01

Lab Name: DDA Line Drawing Algorithm

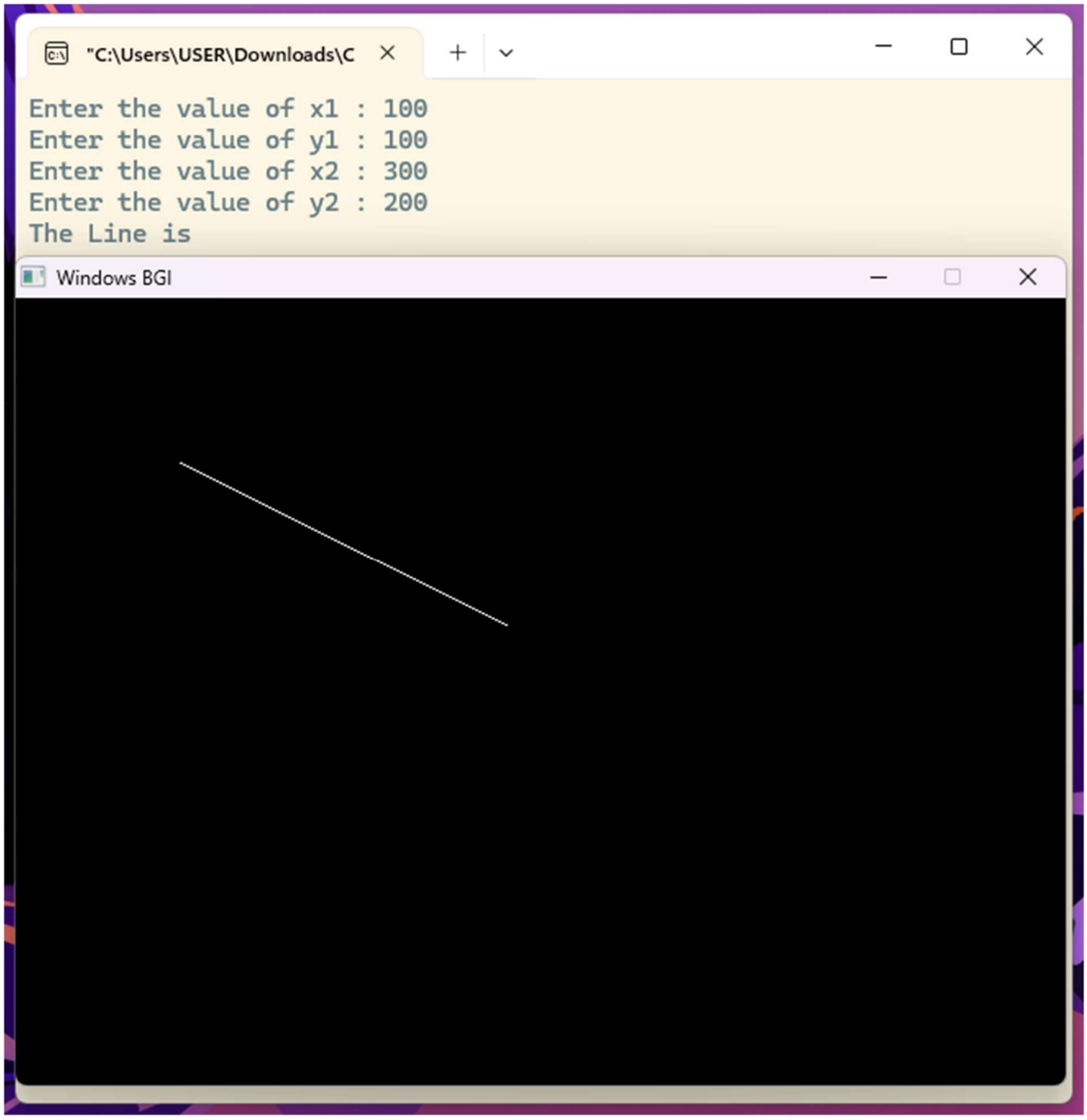
Program:

```
#include <graphics.h>
#include <stdio.h>
#include <math.h>
int main( )
{
    float x,y,x1,y1,x2,y2,dx,dy,pixel;
    int i,gd,gm;

    printf("Enter the value of x1 : ");
    scanf("%f",&x1);
    printf("Enter the value of y1 : ");
    scanf("%f",&y1);
    printf("Enter the value of x2 : ");
    scanf("%f",&x2);
    printf("Enter the value of y2 : ");
    scanf("%f",&y2);
    printf("The Line is\n");
    detectgraph(&gd,&gm);
    initgraph(&gd,&gm,"");

    dx=abs(x2-x1);
    dy=abs(y2-y1);

    if(dx>=dy)
        pixel=dx;
    else
        pixel=dy;

    dx=dx/pixel;
    dy=dy/pixel;

    x=x1;
    y=y1;
    i=1;

    while(i<=pixel)
    {
          putpixel(x,y,WHITE);
          x=x+dx;
          y=y+dy;
          i=i+1;
          delay(30);
    }

    getch();
}
```

Sample Input:

```
Enter the value of x1 : 100
Enter the value of y1 : 100
Enter the value of x2 : 300
Enter the value of y2 : 200
```

Output:



Lab No: 02

Lab Name: Bresenham Line Drawing Algorithm

Program:

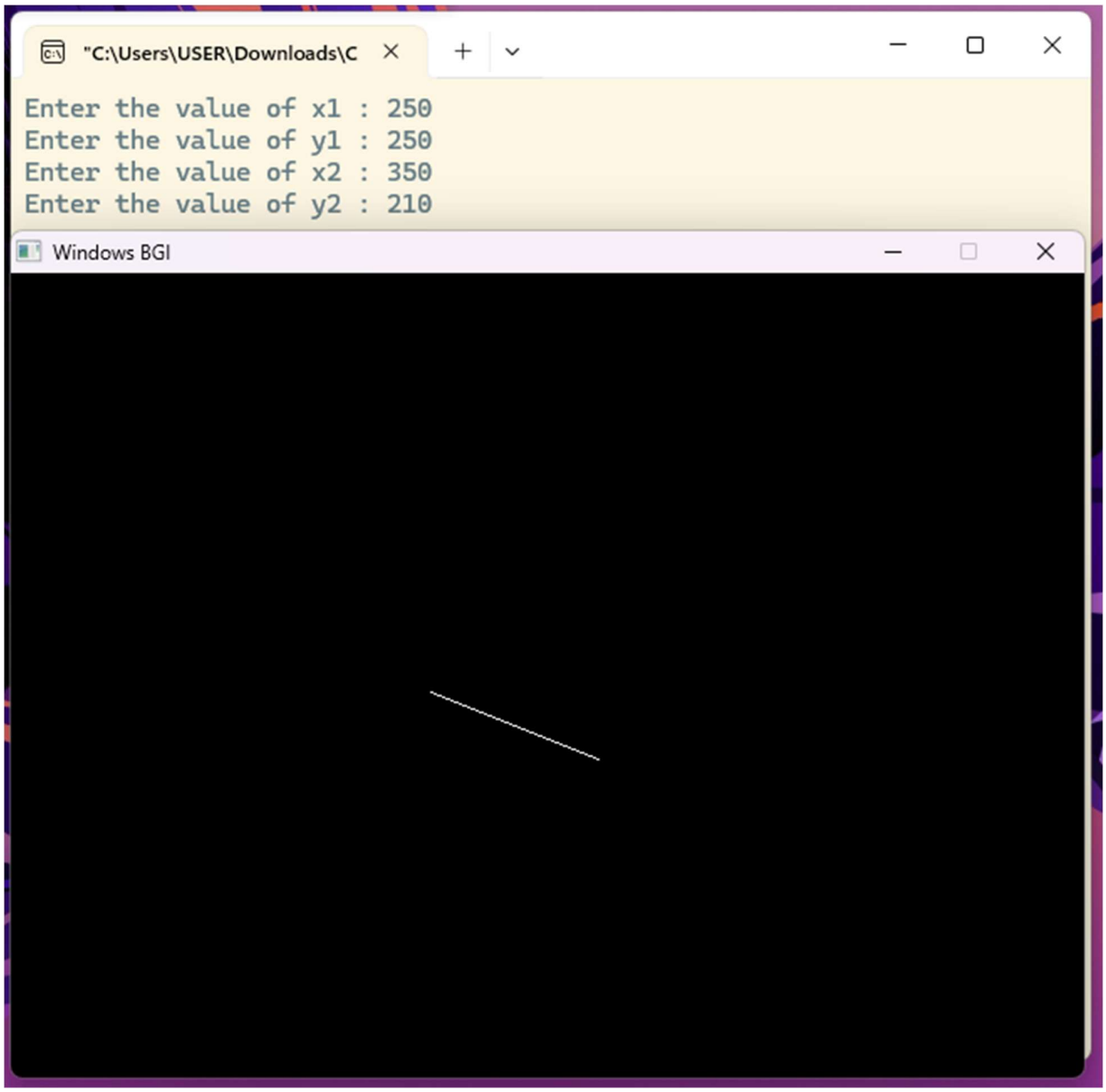
```
#include <graphics.h>
#include <stdio.h>
#include <math.h>

int main()
{
    int x,y,x1,y1,x2,y2,dx,dy,p,gd,gm;

    printf("Enter the value of x1 : ");
    scanf("%d",&x1);
    printf("Enter the value of y1 : ");
    scanf("%d",&y1);
    printf("Enter the value of x2 : ");
    scanf("%d",&x2);
    printf("Enter the value of y2 : ");
    scanf("%d",&y2);

    detectgraph( &gd,&gm);
    initgraph(&gd,&gm,"");

    dx = abs(x2 - x1);
    dy = abs(y2 - y1);

    x = x1;
    y = y1;

    p = 2*dy - dx;

    while(x <= x2)
    {
        putpixel(x,y,WHITE);

        if(p < 0)
        {
            x = x + 1;
            p = p + 2*dy;
        }
        else
        {
            x = x + 1;
            y = y + 1;
            p = p + 2*dy - 2*dx;
        }
        delay(30);
    }
    getch();
```

```
    closegraph();
    return 0;
}
```

Sample Input:

```
Enter the value of x1 : 100
Enter the value of y1 : 100
Enter the value of x2 : 300
Enter the value of y2 : 200
```

Output:



Lab No: 03

Lab Name: Bresenham Circle Drawing Algorithm

Program:

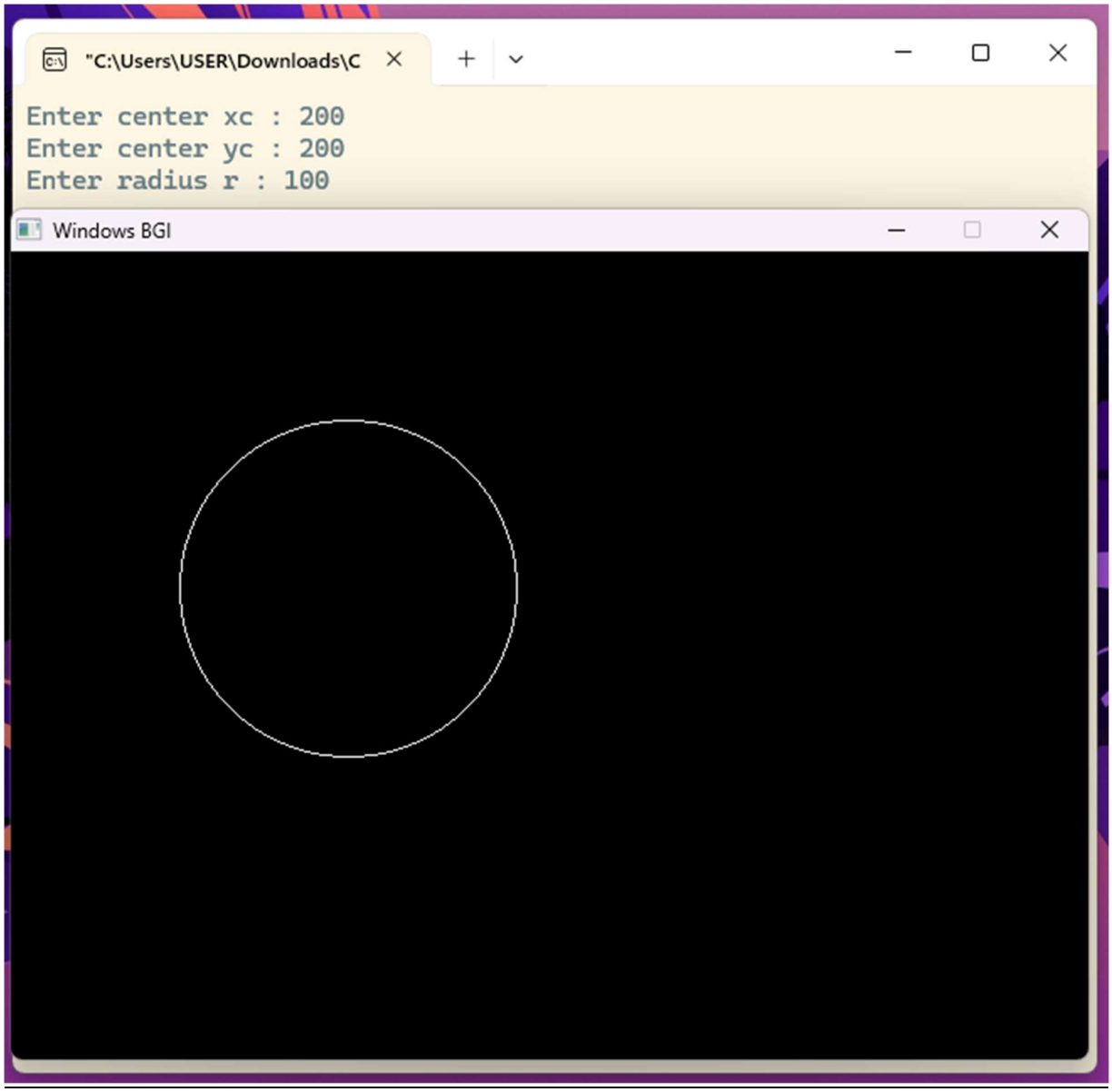
```
#include <graphics.h>
#include <stdio.h>

int main()
{
    int xc,yc,r;
    int x,y,p;
    int gd,gm;

    printf("Enter center xc : ");
    scanf("%d",&xc);
    printf("Enter center yc : ");
    scanf("%d",&yc);
    printf("Enter radius r : ");
    scanf("%d",&r);

    detectgraph(&gd,&gm);
    initgraph(&gd,&gm,"");

    x = 0;
    y = r;
    p = 3 - 2*r;

    while(x <= y)
    {
        putpixel(xc+x,yc+y,WHITE);
        putpixel(xc-x,yc+y,WHITE);
        putpixel(xc+x,yc-y,WHITE);
        putpixel(xc-x,yc-y,WHITE);

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

        if(p < 0)
        {
            p = p + 4*x + 6;
        }
        else
        {
            p = p + 4*(x - y) + 10;
            y = y - 1;
        }

        x = x + 1;
```

```
        delay(30);
    }

    getch();
    closegraph();
    return 0;
}
```

Sample Input:

```
Enter center xc : 200
Enter center yc : 200
Enter radius r : 100
```

Output:



Lab No: 04

Lab Name: Midpoint Circle Drawing Algorithm

Program:

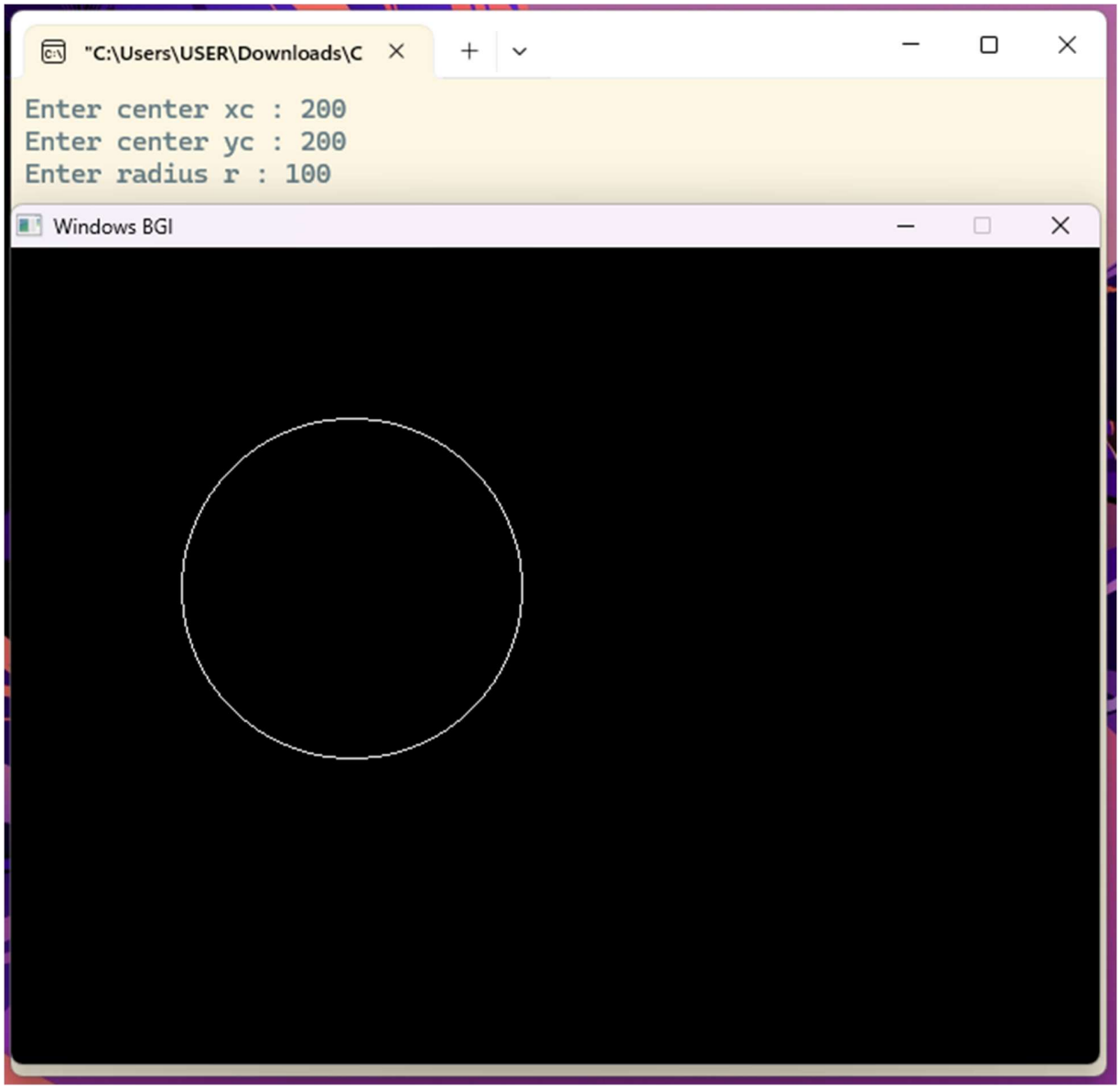
```
#include <graphics.h>
#include <stdio.h>

int main()
{
    int xc,yc,r;
    int x,y,p;
    int gd,gm;

    printf("Enter center xc : ");
    scanf("%d",&xc);
    printf("Enter center yc : ");
    scanf("%d",&yc);
    printf("Enter radius r : ");
    scanf("%d",&r);

    detectgraph(&gd,&gm);
    initgraph(&gd,&gm,"");

    x = 0;
    y = r;
    p = 1 - r;

    while(x <= y)
    {
        putpixel(xc+x,yc+y,WHITE);
        putpixel(xc-x,yc+y,WHITE);
        putpixel(xc+x,yc-y,WHITE);
        putpixel(xc-x,yc-y,WHITE);

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

        if(p < 0)
        {
            p = p + 2*x + 3;
        }
        else
        {
            p = p + 2*(x - y) + 5;
            y = y - 1;
        }

        x = x + 1;
```

```
        delay(30);
    }

    getch();
    closegraph();
    return 0;
}
```

Sample Input:

```
Enter center xc : 200
Enter center yc : 200
Enter radius r : 100
```

Output:



Lab No: 05

Lab Name: Midpoint Ellipse Drawing Algorithm

Program:

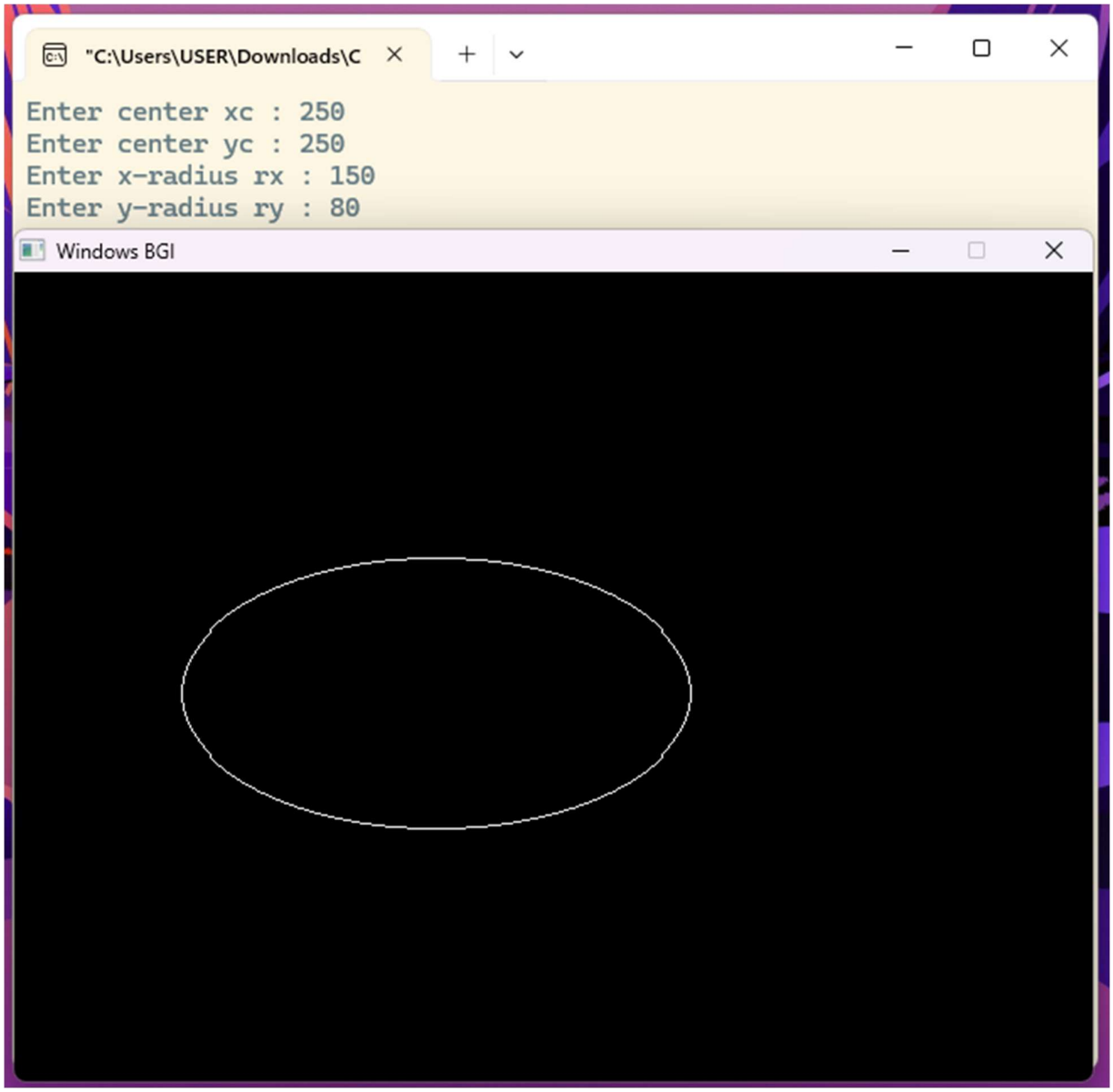
```
#include <graphics.h>
#include <stdio.h>

int main()
{
    int xc,yc;
    int rx,ry;
    int x,y;
    long p1,p2;
    int gd,gm;
    printf("Enter center xc : ");
    scanf("%d",&xc);
    printf("Enter center yc : ");
    scanf("%d",&yc);
    printf("Enter x-radius rx : ");
    scanf("%d",&rx);
    printf("Enter y-radius ry : ");
    scanf("%d",&ry);

    detectgraph(&gd,&gm);
    initgraph(&gd,&gm,"");

    x = 0;
    y = ry;

    p1 = ry*ry - rx*rx*ry + (rx*rx)/4;

    while(2*ry*ry*x <= 2*rx*rx*y)
    {
        putpixel(xc+x,yc+y,WHITE);
        putpixel(xc-x,yc+y,WHITE);
        putpixel(xc+x,yc-y,WHITE);
        putpixel(xc-x,yc-y,WHITE);

        if(p1 < 0)
        {
            p1 = p1 + 2*ry*ry*x + ry*ry;
        }
        else
        {
            p1 = p1 + 2*ry*ry*x - 2*rx*rx*y + ry*ry;
            y = y - 1;
        }
        x = x + 1;
        delay(20);
    }
```

```
    p2 = ry*ry*(x+0.5)*(x+0.5) + rx*rx*(y-1)*(y-1) -
                rx*rx*ry*ry;

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

        if(p2 > 0)
        {
            p2 = p2 - 2*rx*rx*y + rx*rx;
        }
        else
        {
            p2 = p2 + 2*ry*ry*x - 2*rx*rx*y + rx*rx;
            x = x + 1;
        }
        y = y - 1;
        delay(20);
    }

    getch();
    closegraph();
    return 0;
}
```

Sample Input:

```
Enter center xc : 250
Enter center yc : 250
Enter x-radius rx : 150
Enter y-radius ry : 80
```

Output:


"C:\Users\USER\Downloads\C
Enter center xc : 250
Enter center yc : 250
Enter x-radius rx : 150
Enter y-radius ry : 80
Windows BGI

Lab No: 06

Lab Name: Liang–Barsky Line Clipping Algorithm

Program:

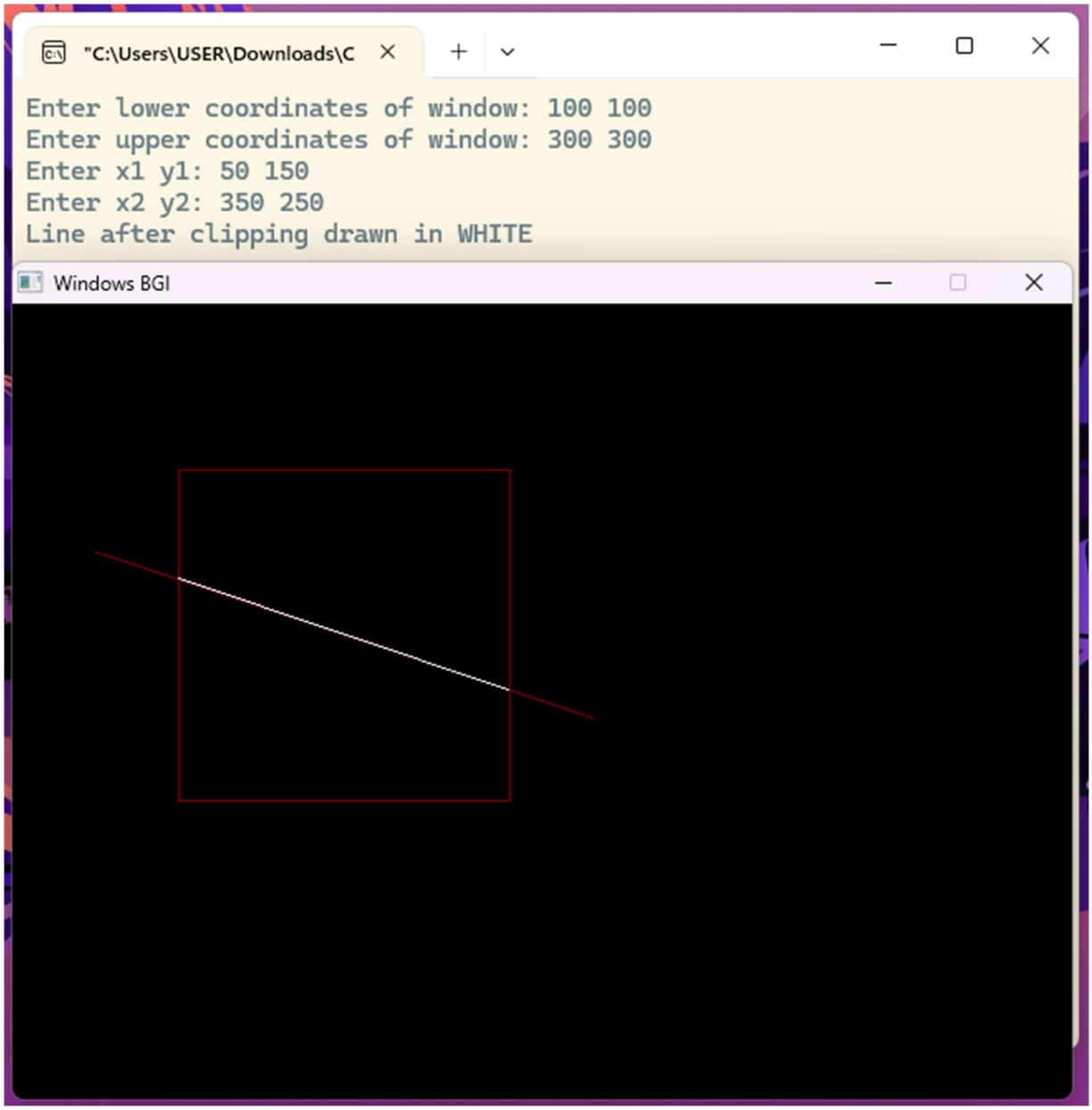
```
#include<stdio.h>
#include<graphics.h>

int main()
{
    int gd=DETECT, gm;
    int x1, y1, x2, y2;
    int xmin, ymin, xmax, ymax;
    int xx1, yy1, xx2, yy2;
    int dx, dy, i;
    int p[4], q[4];
    float t[4], t1, t2;
    printf("Enter lower coordinates of window: ");
    scanf("%d %d", &xmin, &ymin);
    printf("Enter upper coordinates of window: ");
    scanf("%d %d", &xmax, &ymax);
    printf("Enter x1 y1: ");
    scanf("%d %d", &x1, &y1);
    printf("Enter x2 y2: ");
    scanf("%d %d", &x2, &y2);

    initgraph(&gd,&gm," ");

    setcolor(RED);
    rectangle(xmin, ymin, xmax, ymax);
    line(x1, y1, x2, y2);

    dx = x2 - x1;
    dy = y2 - y1;

    p[0] = -dx; q[0] = x1 - xmin;
    p[1] =  dx; q[1] = xmax - x1;
    p[2] = -dy; q[2] = y1 - ymin;
    p[3] =  dy; q[3] = ymax - y1;

    for(i = 0; i < 4; i++)
    {
        if(p[i] != 0)
            t[i] = (float)q[i] / p[i];
        else if(p[i] == 0 && q[i] < 0)
        {
            printf("Line completely outside the window\n");
            getch();
            closegraph();
            return 0;
        }
```

```
        else
            t[i] = (i < 2) ? 0.0 : 1.0; // partially inside
    }

    t1 = (t[0] > t[2]) ? t[0] : t[2];
    t2 = (t[1] < t[3]) ? t[1] : t[3];

    if(t1 < t2)
    {
        xx1 = x1 + t1*dx;
        yy1 = y1 + t1*dy;
        xx2 = x1 + t2*dx;
        yy2 = y1 + t2*dy;
        printf("Line after clipping drawn in WHITE\n");
        setcolor(WHITE);
        line(xx1, yy1, xx2, yy2);
    }
    else
    {
        printf("Line lies completely outside the window\n");
    }

    getch();
    closegraph();
    return 0;
}
```

Sample Input:

```
Enter lower coordinates of window: 100 100
Enter upper coordinates of window: 300 300
Enter x1 y1: 50 150
Enter x2 y2: 350 250
```

Output:


"C:\Users\USER\Downloads\C
Enter lower coordinates of window: 100 100
Enter upper coordinates of window: 300 300
Enter x1 y1: 50 150
Enter x2 y2: 350 250
Line after clipping drawn in WHITE
Windows BGI