

ATMA RAM SANATAN DHARMA COLLEGE
DELHI UNIVERSITY



COMPUTER GRAPHICS

Assignment



SUBMITTED BY:

DUSHYANT KUMAR SINGH

B.Sc. (Hons) Computer Science)

Roll No – 19/78113

Semester - VI

SUBMITTED TO:

MANISHA BAGRI MA'AM

(DEPARTMENT OF COMPUTER SCIENCE)

Q1. DDA LINE DRAWING ALGORITHM.

```
#include <graphics.h>

#include <iostream.h>

#include <math.h>

#include <dos.h>

void main( )
{
    float x,y,x1,y1,x2,y2,dx,dy,step;
    int i,gd=DETECT,gm;
    initgraph(&gd,&gm,"c:\\\\turbo3\\bgi");
    cout<<"Enter the value of x1 and y1 : ";
    cin>>x1>>y1;
    cout<<"Enter the value of x2 and y2: ";
    cin>>x2>>y2;
    dx=abs(x2-x1);
    dy=abs(y2-y1);
    if(dx>=dy)
        step=dx;
    else
        step=dy;
    dx=dx/step;
    dy=dy/step;
    x=x1;
    y=y1;
    i=1;
    while(i<=step)
```

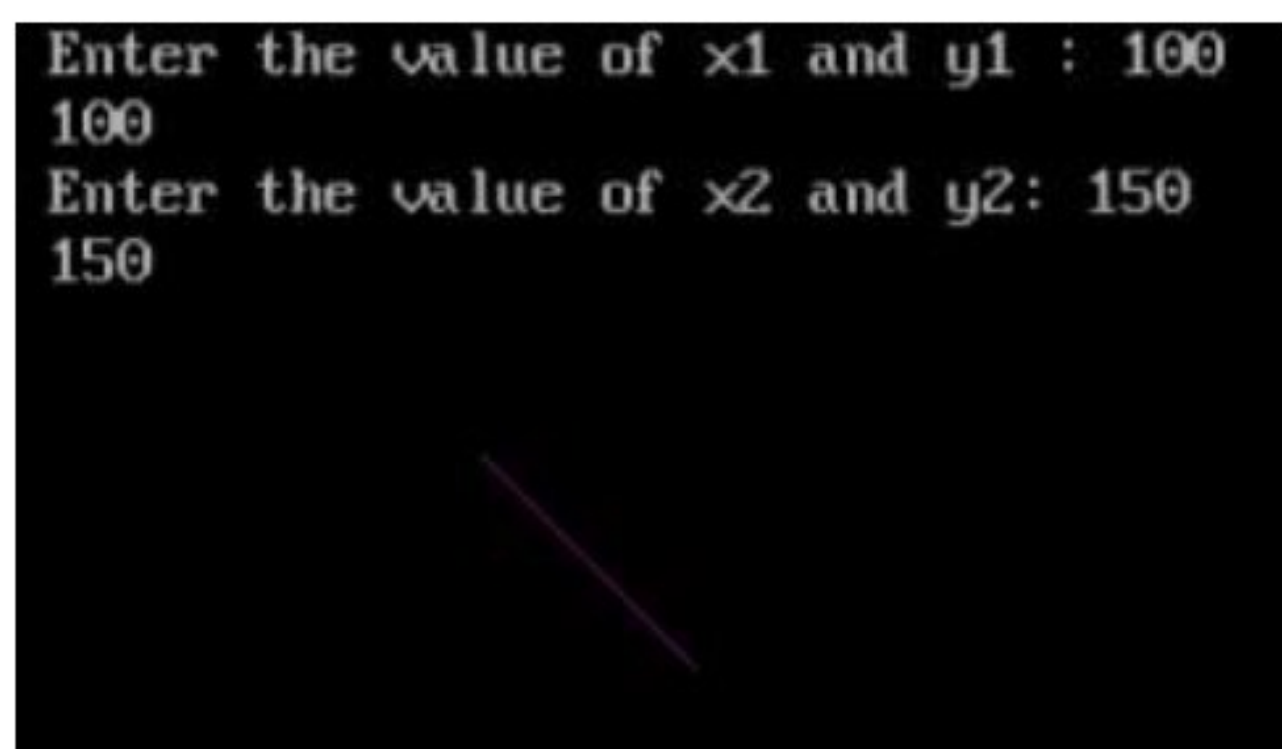
```

    {
        putpixel(x,y,5);
        x=x+dx;
        y=y+dy;
        i=i+1;
        delay(500);
    }

    closegraph();
}

```

OUTPUT:



Q2. BRESENHAM'S LINE ALGORITHM.

```
#include<iostream.h>
```

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

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

```
void drawline(int x0, int y0, int x1, int y1)
```

```
{
```

```
    int dx, dy, p, x, y;
```

```
        dx=x1-x0;
```

```
dy=y1-y0;

x=x0;
y=y0;

p=2*dy-dx;

while(x<x1)
{
    if(p>=0)
    {
        putpixel(x,y,7);
        y=y+1;
        p=p+2*dy-2*dx;
    }
    else
    {
        putpixel(x,y,7);
        p=p+2*dy;
    }
    x=x+1;
}
}
```

```
int main()
{
    int gdriver=DETECT, gmode, error, x0, y0, x1, y1;
    initgraph(&gdriver, &gmode, "c:\\turbo3\\bgi");

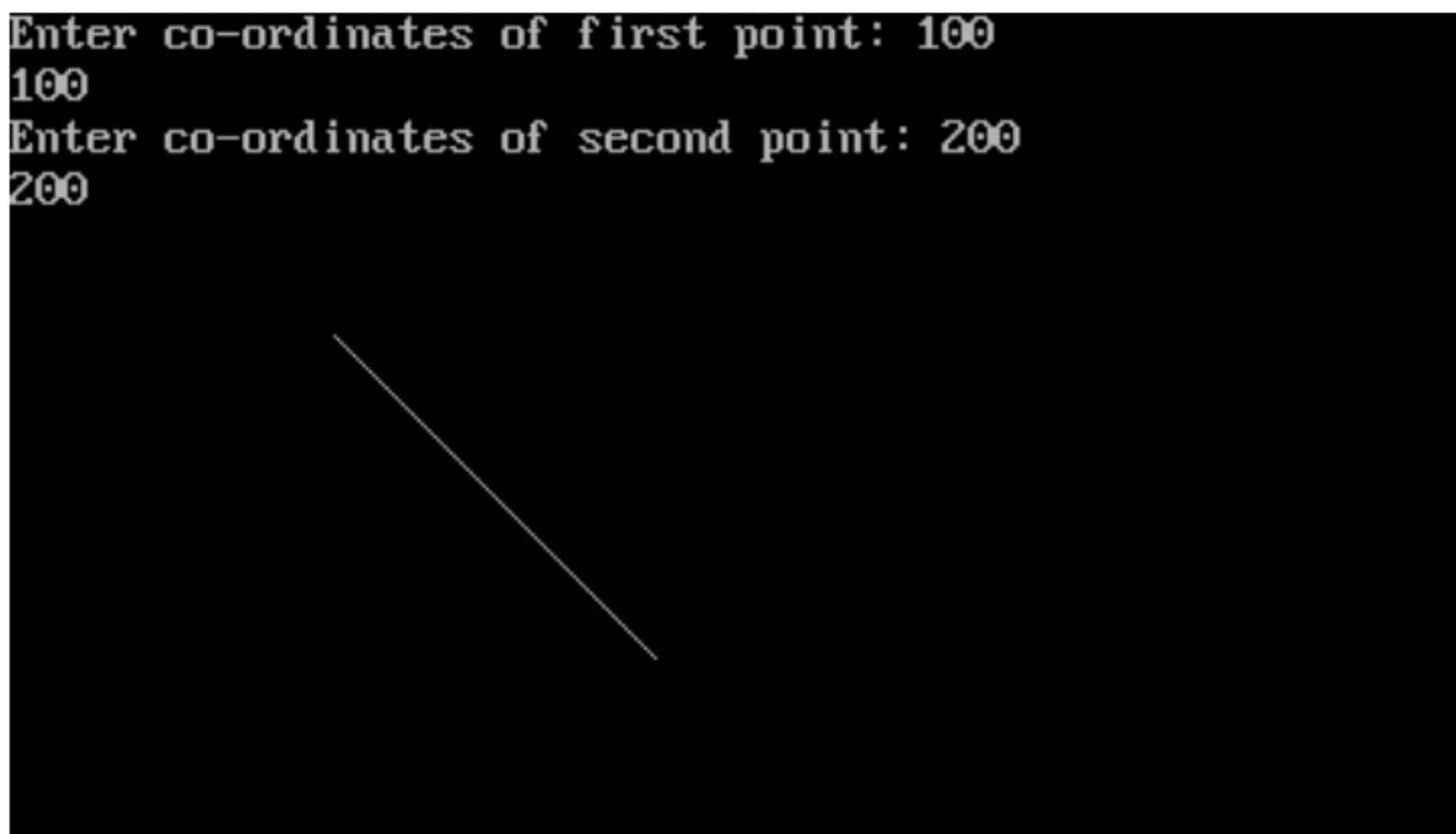
    cout<<"Enter co-ordinates of first point: ";
    cin>>x0>>y0;

    cout<<"Enter co-ordinates of second point: ";
    cin>>x1>>y1;
    drawline(x0, y0, x1, y1);
    delay(500);

    return 0;
}
```

OUTPUT:

```
Enter co-ordinates of first point: 100
100
Enter co-ordinates of second point: 200
200
```



Q3. BRESENHAM'S CIRCLE ALGORITHM.

```
#include<iostream.h>
```

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

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

```
void drawcircle(int x0, int y0, int radius)
```

```
{
```

```
    int x = radius;
```

```
    int y = 0;
```

```
    int err = 0;
```

```
    while (x >= y)
```

```
    {
```

```
        putpixel(x0 + x, y0 + y, 7);
```

```
        putpixel(x0 + y, y0 + x, 7);
```

```
        putpixel(x0 - y, y0 + x, 7);
```

```
        putpixel(x0 - x, y0 + y, 7);
```

```
        putpixel(x0 - x, y0 - y, 7);
```

```
        putpixel(x0 - y, y0 - x, 7);
```

```
        putpixel(x0 + y, y0 - x, 7);
```

```
        putpixel(x0 + x, y0 - y, 7);
```

```
        if (err <= 0)
```

```
        { y += 1;
```

```
            err += 2*y + 1;
```

```
        }
```

```
        if (err > 0)
```



```

        { x -= 1;
          err -= 2*x + 1;
        }
      }
}

int main()
{
    int gdriver=DETECT, gmode, error, x, y, r;
    initgraph(&gdriver, &gmode, "c:\\turbo3\\bgi");
    cout<<"Enter radius of circle: ";
    cin>>r;
    cout<<"Enter co-ordinates of center(x and y): ";
    cin>>x>>y;
    drawcircle(x, y, r);
    delay(500);
    return 0;
}

```

OUTPUT:

```

Enter radius of circle: 100
Enter co-ordinates of center(x and y): 150
150

```

