NAME = SURAJ THAPLIYAL

SET-C

Roll NO. = 1121153

SUBJECT = COMPUTER GRAPHIC

DATE = 16-6-2021

Q1.

Ans. Step1 - Start Algorithm

Step2 : Declare variable $x_1, x_2, y_1, y_2, d, i_1, i_2, dx, dy$

Step3 : Enter value of x_1, y_1, x_2, y_2 Where x_1, y_1 are coordinates of starting point And x_2, y_2 are coordinates of Ending point

Step4 : Calculate $dx = x_2 - x_1$

Calculate $dy = y_2 - y_1$

Calculate $i_1 = 2*dy$

Calculate $i_2 = 2*(dy - dx)$

Calculate $d = i_1 - dx$

Step5 : Consider (x, y) as starting point and x_{end} as maximum possible value of x.

If $dx < 0$

Then $x = x_2$

$y = y_2$

$x_{end} = x_1$

If $dx > 0$

Then $x = x_1$

$y = y_1$

$x_{end} = x_2$

Step6 : Generate point at (x, y) coordinates.

Step 7. Check if whole line is generated

If $x >= x_{end}$

Stop.

Step 8: Calculate co-ordinates of the next pixel

If $d < 0$

Then $d = d + i_1$

If $d \geq 0$

Then $d = d + i_2$

Increment $y = y + 1$

Step 9: Increment $x = x + 1$

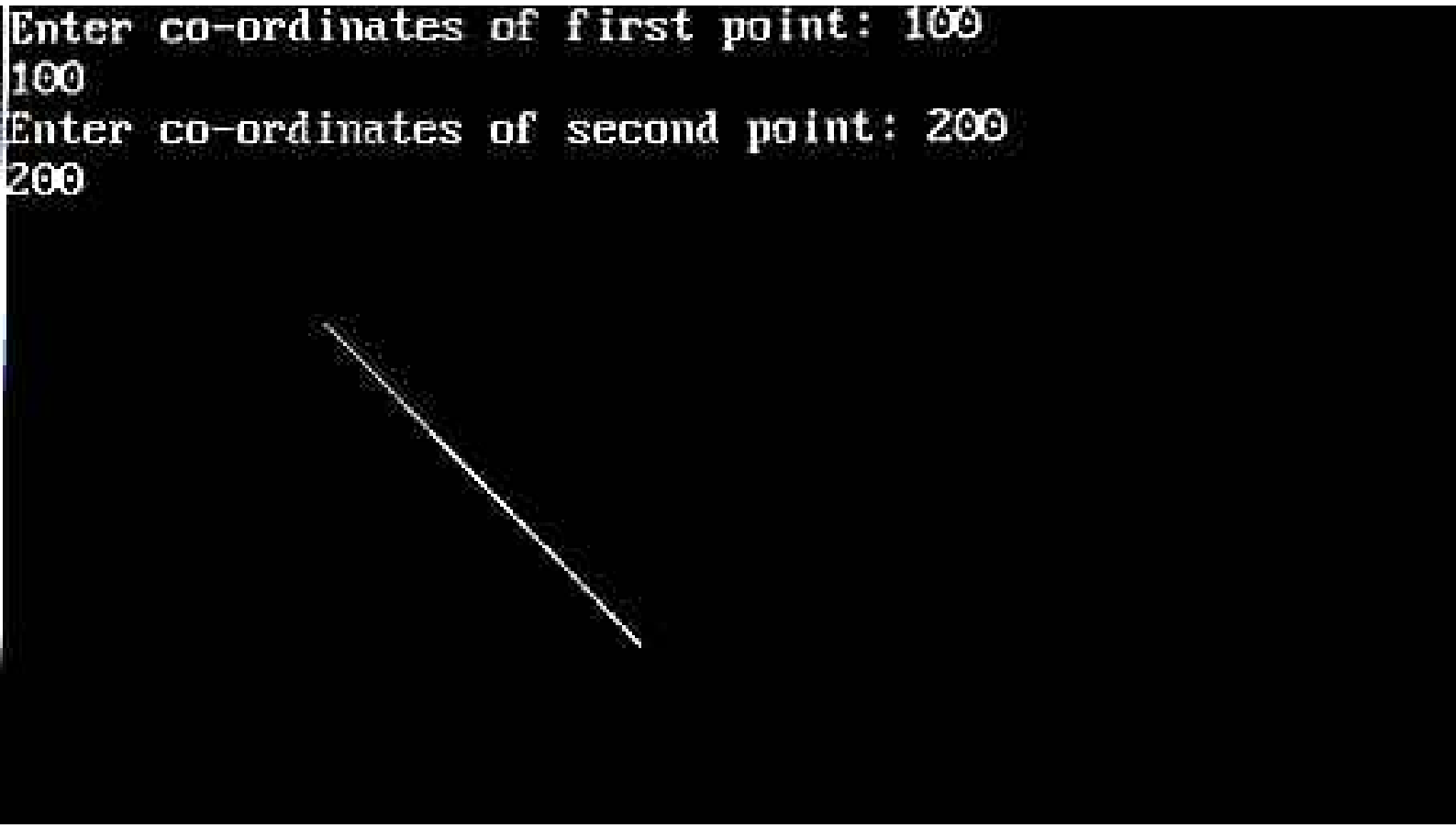
Step 10: Draw a point of latest (x, y) coordinates

Step 11: Go to step 7

Step 12: End of Algorithm

```
#include <stdio.h>
#include <graphics.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 gdrive=DETECT, gmode, error, x0, y0, x1, y1;
  initgraph(&gdrive, &gmode, "C:\\turboc3\\bgi");
  printf("Enter co-ordinates of first point:");
 scanf("%d%d", &x0, &y0);
printf("Enter co-ordinates of second point:");
scanf("%d%d", &x1, &y1);
drawline(x0, y0, x1, y1);
 return 0;
}
```


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

NAME = SURAJ THAPLIYAL

ROLL NO = 112153

SUBJECT = COMPUTER GRAPHICS

DATE = 16-6-2021

Q2.

Ans. Algorithm for Midpoint Circle.

Step 1. Get radius and coordinates from the user.

Step 2. Find out the decision parameter that decides the nearest point to select using:

$$d = 5/4 - r$$

Step 3. While Y is greater than X do

- if d is smaller than 0, then
  $y = y$
  $n = n + 1$
  $d = 2x + 1$
- else
  $y = y - 1$
  $x = x + 1$
  $d = d + 2x - 2y + 1$

Step 4. Determine and plot the symmetry point for all eight octants.

Step 5. Repeat step 3 and 4, till $y > n$

```
#include <stdio.h>
#include <graphic.h>
#include <conio.h>
void main()
{
  int x,y,x_mid,y_mid,radius,dp;
   int g_mode, g_driver = DETECT;
   clrscr();
   initgraph (&g_driver, &g_mode, "C:\\TURROC3\\BGI");
  printf ("***** MID POINT circle drawing algorithm *****\n\n");
   printf ("\nenter the coordinates = ");
  scanf ("%d%d", &x_mid, &y_mid);
  printf ("\n now enter the radius = ");
   scanf ("%d", &radius);
     x = 0;
     y = radius;
    dp = 1 - radius;
    do
    {
     putpixel (x_mid+x, y_mid+y, YELLOW);
     putpixel (x_mid+y, y_mid+x, YELLOW);
     putpixel (x_mid-y, y_mid+x, YELLOW);
    putpixel (x_mid-x, y_mid+y, YELLOW);
    put pixel (x_mid-x, y_mid-y, YELLOW);
    putpixel (x_mid-Y, y_mid-x, YELLOW);
    putpixel (x_mid+Y, y_mid-x, YELLOW);
    putpixel (x_mid+x, y_mid-y, YELLOW);
    if (dp<0) {
      dp+= (2*x)+1;
    }
```

```
    else{
    y=y-1;
    dp+=(2*x)-(2*y)+1;
    }
    x=x+1;
  }while(y>x);
   getch();
}
```

********** MID POINT Circle drawing algorithm *********

enter the coordinates= 250 250

now enter the radius =100