

## End-Term

Name - Jasmeet Kaur

Course - BCA 'B' 'C'

Rollno - 10

University Rollno - 1171067

Paper Code - PBC 602

Subject - Computer Graphics

### Answers

#### 1 Source Code

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

```
void main ( )
```

```
{ float x, y, x1, y1, x2, y2, dx, dy, steps, p;
```

```
int i=1; gd = DETECT, gm;
```

```
printf("Enter (x1, y1):");
```

```
scanf("%f%f", &x1, &y1);
```

```
printf("Enter (x2, y2):");
```

```
scanf("%f%f", &x2, &y2);
```

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

```
dx = x2 - x1;
```

```
dy = y2 - y1;
```

```
steps = dx - 1;
```

```
int pk = (2 * dy) - dx;
```

```
p = pk;
```

```
x = x1;
```

```
y = y1;
```

```
while (i <= steps)
```

```
{ if (p < 0)
```

```
{ putpixel (x, y, BLUE);
```

```
x = x + 1;
```

```
y = y;
```



```

    p = p + (2 * dy);
    delay(50);
}
else {
    putpixel(x, y, BLUE);
    x = x + 1;
    y = y + 1;
    p = p + (2 * dy) - (2 * dx);
    delay(50);
}
i++;
}
getch();
closegraph();
}

```

## Algorithm

- Step 1: Start Algorithm
- Step 2: Declare variable  $x, y, x_1, y_1, x_2, y_2, dx, dy, i$
- Step 3: Enter Value of  $x_1, y_1, x_2, y_2$   
 where  $x_1, y_1$  are Co-ordinates of starting point  
 And  $x_2, y_2$  are Co-ordinates of ending points.
- Step 4: Calculate  $dx = x_2 - x_1$   
 Calculate  $dy = y_2 - y_1$   
 Calculate  $p_2 = 2 * (dy - dx)$  &  $d = p_1 - dx$
- Step 5: 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  $x = x_1$

$y = y_1$ ,  $x_{end} = x_2$

Step 6 - Generate point at  $(x, y)$  Coordinates

if  $x \geq x_{end}$

Stop

Step 8 - Calculate co-ordinates of the next pixel

if  $d < 0$

then  $d = d + 1$

if  $d > 0$ , then  $d = d + 2$

Increment  $y = y + 1$

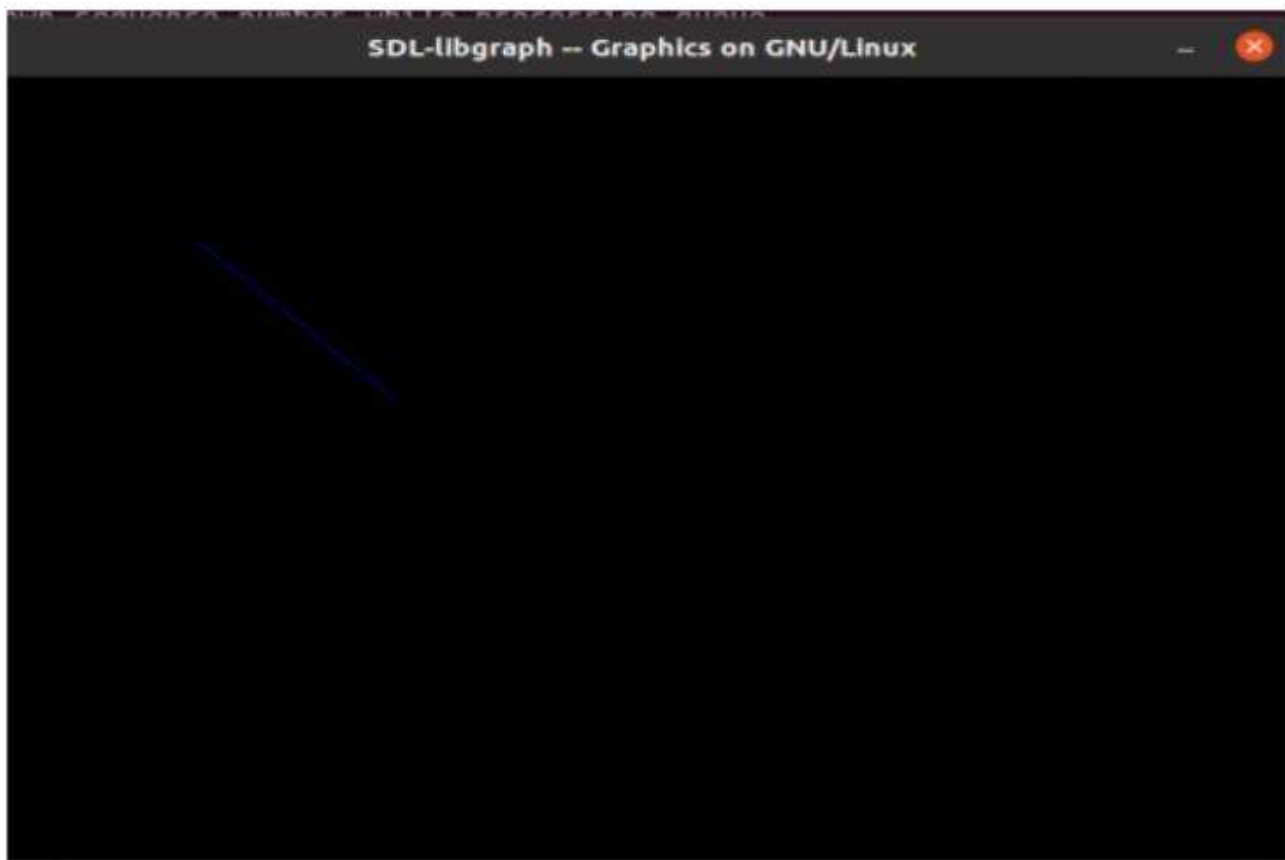
Step 9 - Increment  $x = x + 1$

Step 10 - Draw point of latest  $(x, y)$  coordinates

Step 11 - Go to step 7

Step 12 - End of Algorithm

## OUTPUT:





Name - Jasmeeet Kaur

Rollno - 10

Paper Code - PBC 602

Course - BCA - 6 (C)

University Rollno - 1121067

Subject = Computer Graphics

Answers.

Q Source Code

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

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

```
int main()
```

```
{ int gd = DETECT, gm;
```

```
int x, y, p, xc = 200, yc = 200;
```

```
printf("Enter radius");
```

```
scanf("%d", &x);
```

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

```
x = 0;
```

```
y = x;
```

```
p = 1 - x;
```

```
for (x = 0; x <= y; x++)
```

```
{ if (p < 0)
```

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

```
p = p + (2 * x) + 1;
```

```
}
```

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

```
p = p + (2 * x) - (2 * y) + 1;
```

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

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



```

putpixel(xc - x, yc + y, 7);
putpixel(xc - y, yc - y, 7);
putpixel(xc - y, yc + x, 7);
putpixel(xc + x, yc - y, 7);
}
getch();
closegraph();
return 0;
}

```

## Algorithm

Step 1:- Start Algorithm.

Step 2:- Plot the Center Coordinates  $(p_0, q_0)$  follows -  
 $p_0 = 0, q_0 = r$

Step 3:- Now, calculate the int decision parameter  $d_0 = 1 - r$

Step 4. Assume the starting coordinates  $(p_k, q_k)$   
 The next coordinates will be  $(p_{k+1}, q_{k+1})$

Find the next point of first octant according to  $d_k$ .

Step 5:- follows these 2 Cases -

Case 1: If  $d_k < 0$ , then

$$p_{k+1} = p_k + 1$$

$$q_{k+1} = q_k$$

$$d_{k+1} = d_k + 2p_{k+1} + 1$$

Case 2: If  $d_k \geq 0$ , then

$$p_{k+1} = p_k + 1$$

$$q_{k+1} = q_k - 1$$

$$d_{k+1} = d_k - 2(q_{k+1} + p_{k+1}) + 1$$

Step 6- If center not  $(0, 0)$  points will be

$$x_{\text{Coordinate}} = x_c + p_0$$

$$y_{\text{Coordinate}} = y_c + q_0$$

Step 7- Repeat Steps 5 & 6 until  $x \geq y$

Step 8- Stop

**OUTPUT :**

