Name- Vishu Negi Roll No- 1121163 (48)

Paper Code → TBC-602 Course- BCA 6C

Program

```
#include<stdio.h>
#include<graphics.h>
int main()
{
int main()
{ int rou(float num)
{
return num<0 ? num-0.5 : num+0.5;
}
int x1 = 100, x2 = 300, y1 =100, y2=200;
int gd = DETECT, gm;
float pk, pkk, x, y, step;
int dx= x2-x1;
int dy = y2-y1;
pk = 2* dx-dy;
if (dx>dy)
step = dx;
else
step =dy;
initgraph (&gd, &gm, "");
outtextxy (x1, y1, "A");
outtextxy (x2, y2, "B");
putpixel (x1, y1, WHITE);
x=x1, y=y1;
while (step>0)
{
if (pk<0)
{
pkk = pk + 2* dy;
} else
```

```
{
pkk = pk + 2* dy - 2*dy;
y++;
}
putpixel (rou(x), rou(y), WHITE);
x++;
step--;
}
getch();
return 0;
}
```

Algorithm

Step 1 : Start Algorithm

Step 2 : Declare variable x1, x2, y1, d, i1, i2, dx, dy

Step 3 : Enter Value of x1, y1, x2, y2 where x1, y1 are coordinates of starting point
And x2, y2 are coordinates of ending point

Step 4 : Calculate dx = x2 - x1
Calculate dy = y2 - y1
Calculate i1 = 2*dy
Calculate i2 = 2*(dy - dx)
Calculate d = i1 - dx

Step 5 : Consider (x, y) as starting point and xend as maximum possible value of x

If dx < 0
Then x = x2
y = y2
xend = x1

If dx > 0
then x = x1
y = y1
xend = x2

Step 6: 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 $d = d + i1$

If $d \geq 0$

Then $d = d + i2$

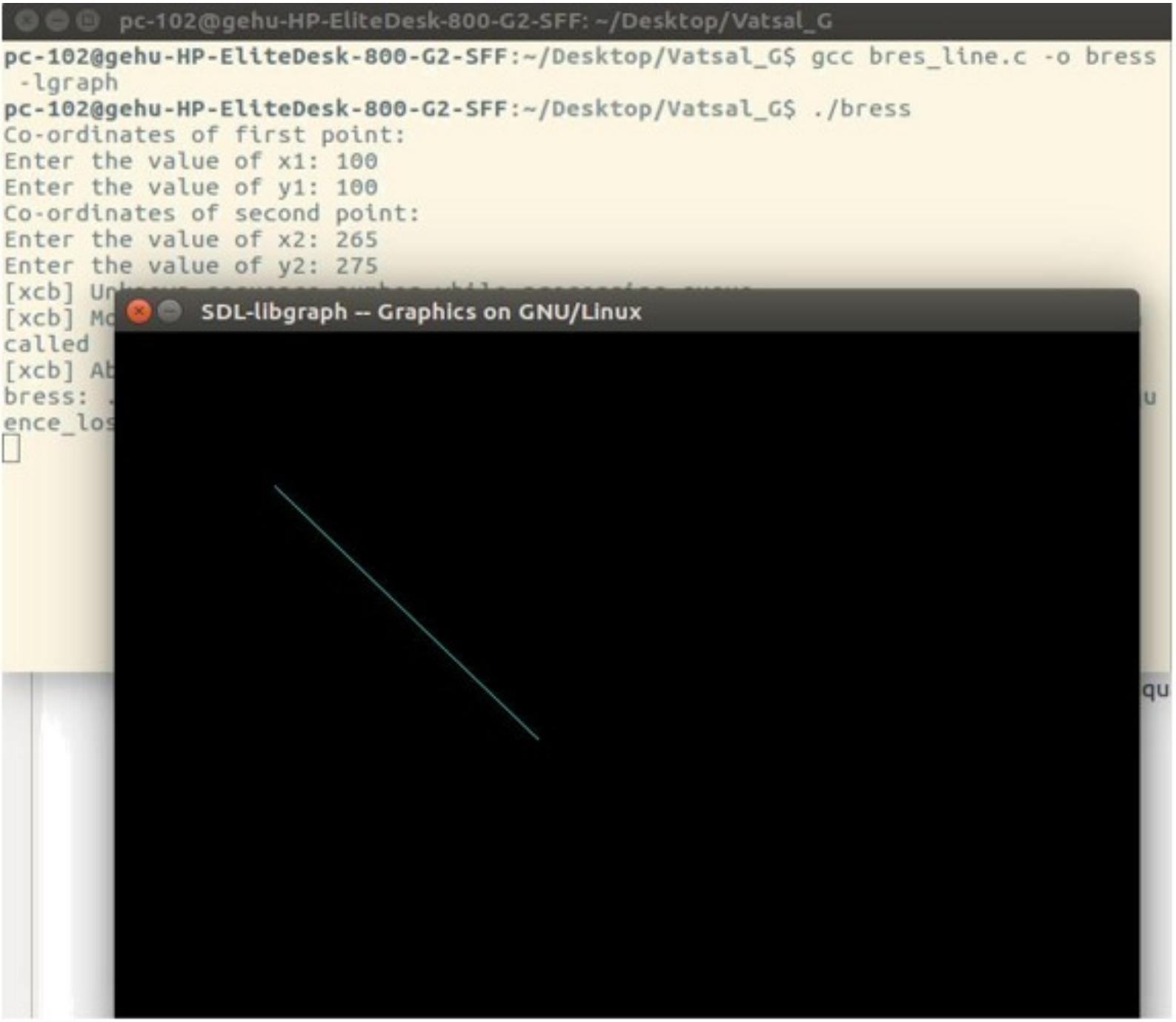
Increment $y = y + 1$

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


pc-102@gehu-HP-EliteDesk-800-G2-SFF: ~/Desktop/Vatsal_G
pc-102@gehu-HP-EliteDesk-800-G2-SFF:~/Desktop/Vatsal_G$ gcc bres_line.c -o bress
-lgraph
pc-102@gehu-HP-EliteDesk-800-G2-SFF:~/Desktop/Vatsal_G$ ./bress
Co-ordinates of first point:
Enter the value of x1: 100
Enter the value of y1: 100
Co-ordinates of second point:
Enter the value of x2: 265
Enter the value of y2: 275
SDL-libgraph -- Graphics on GNU/Linux

Name - Vibhu Negi Roll No - 1121163 (48)

Paper Code : TBC-602 Course - BCA 6 C

Ans 2

```
#include <stdio.h>
# include <graphics.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;
    printf("Enter radius of circle:");
    scanf("%d", &r);
    printf("Enter co-ordinates of center(x and y):");
    scanf("%d%d", &x, &y);
    initgraph(&gdriver, &gmode, "");
    drawcircle(x, y, r);
    delay(9999999);
    return 0;
}
```

Algorithm

Step 1: Assign the starting point coordinates (X_0, Y_0) as-

- $X_0 = 0$
- $Y_0 = R$

Step 2: Calculate the value of intial decesion parameter P_0 as-

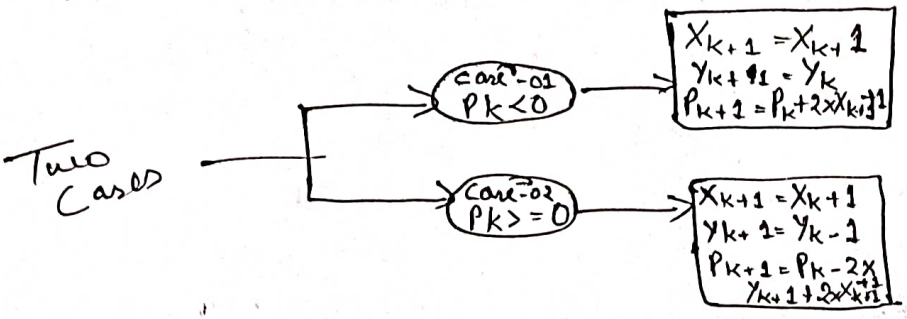
$$P_0 = 1 - R$$

Step-3

Suppose the Current point is (X_k, Y_k) and the next point is (X_{k+1}, Y_{k+1})

Find the next point of the first octant depending on the Value of decision parameter P_k

Follow the below two cases



Step-4

In the given centre point (X_0, Y_0) is not $(0,0)$, then do the following and plot the point

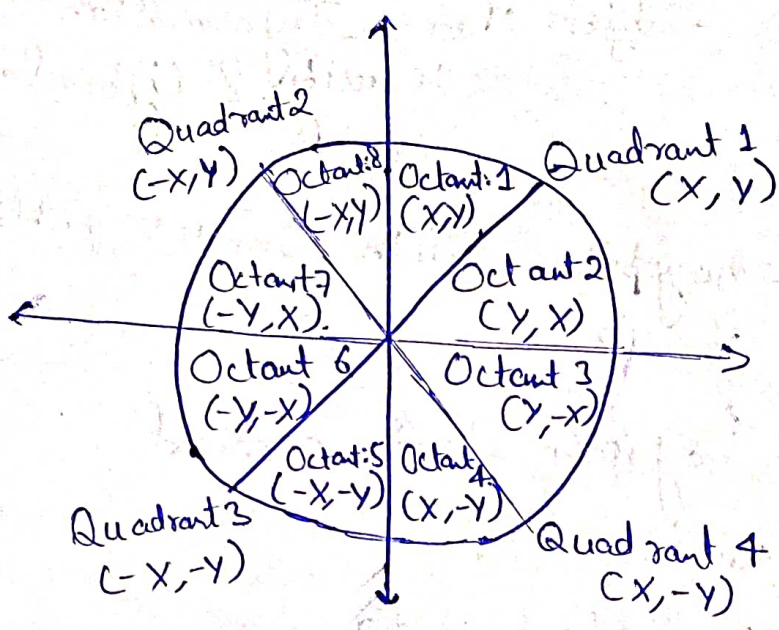
- $X_{plot} = X_c + X_0$
- $Y_{plot} = Y_c + Y_0$

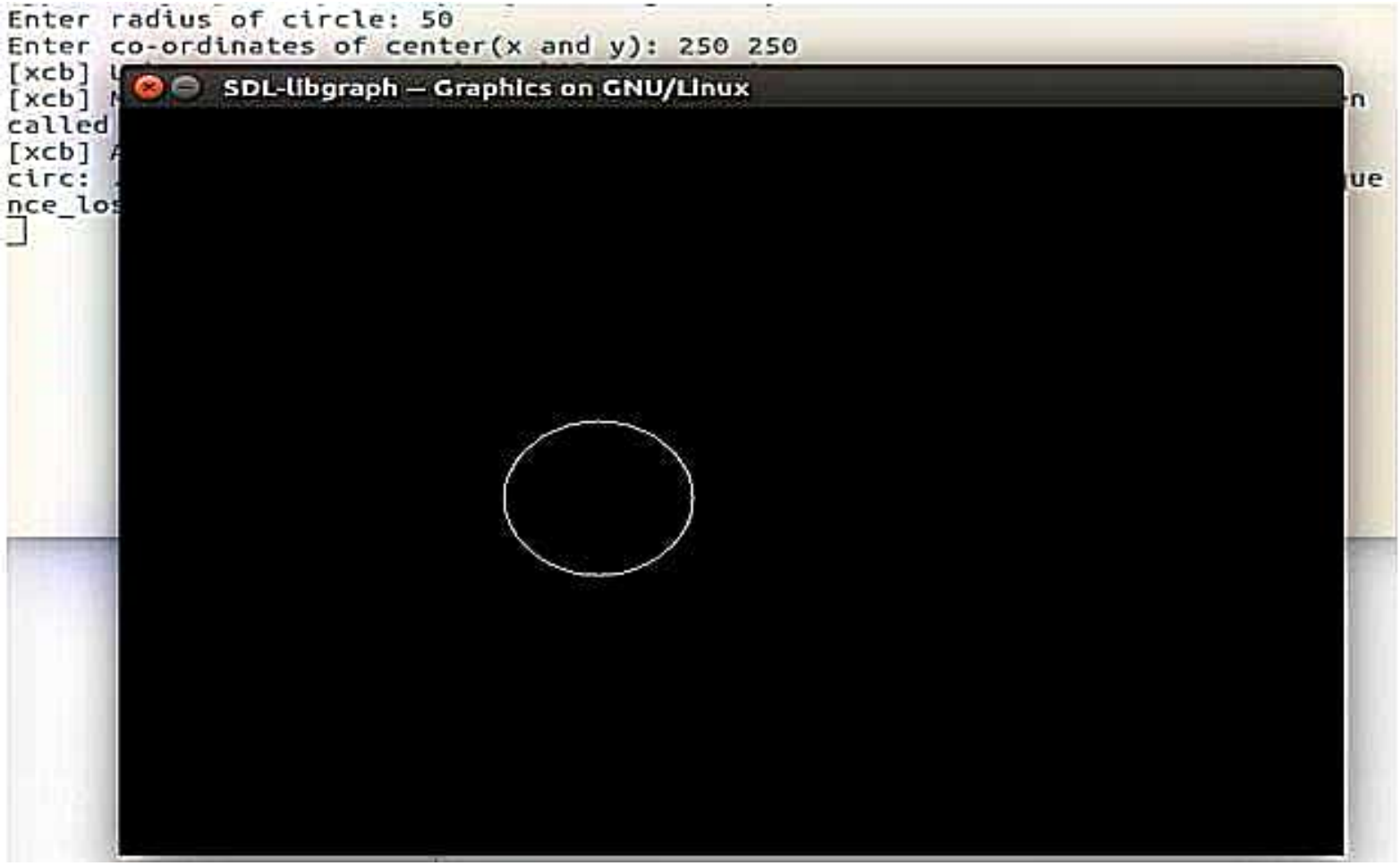
Here, (X_c, Y_c) denotes the current value of X and Y Coordinates.

Step5:

Keep repeating Step-03 and Step-04 Until $X_{plot} >= Y_{plot}$.

Step 6: generates all the points for one octant To find the points for other seven octants, follow the eight Symmetry property of Circle


Quadrant 2
(-X, Y)
Octant: 8
(-X, Y)
Octant: 1
(X, Y)
Quadrant 1
(X, Y)
Octant 7
(-Y, X)
Octant 2
(Y, X)
Octant 6
(-Y, -X)
Octant 3
(Y, -X)
Octant: 5
(-X, -Y)
Octant 4
(X, -Y)
Quadrant 3
(-X, -Y)
Quadrant 4
(X, -Y)


Enter radius of circle: 50
Enter co-ordinates of center(x and y): 250 250
SDL-libgraph – Graphics on GNU/Linux