Name :- Himanshu Gusain   Course :- BCA '6C'

Roll no. :- 05   Subject :- Computer Graphics

Que 1 :- • Bresenham's line drawing algorithm :-

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

Scanned By ScanIt

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

Algorithm :-

Step 1 :- Start Algorithm

Step 2 :- Declare variable $x_1, x_2, y_1, y_2, d, i_1, i_2, dx, dy$

Step 3 :- 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

Step 4 :- Calculate $dx = x_2 - x_1$
calculate $dy = y_2 - y_1$
calculate $i_1 = 2* dy$
calculate $i_2 = 2*(dy - dx)$
calculate $d = i_1 - dx$

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

```
if dx < 0
  Then x = x2
  y = y2
  xend = x1
if dx > 0
  Then x = x1
  y = y1
  xend = x2
```

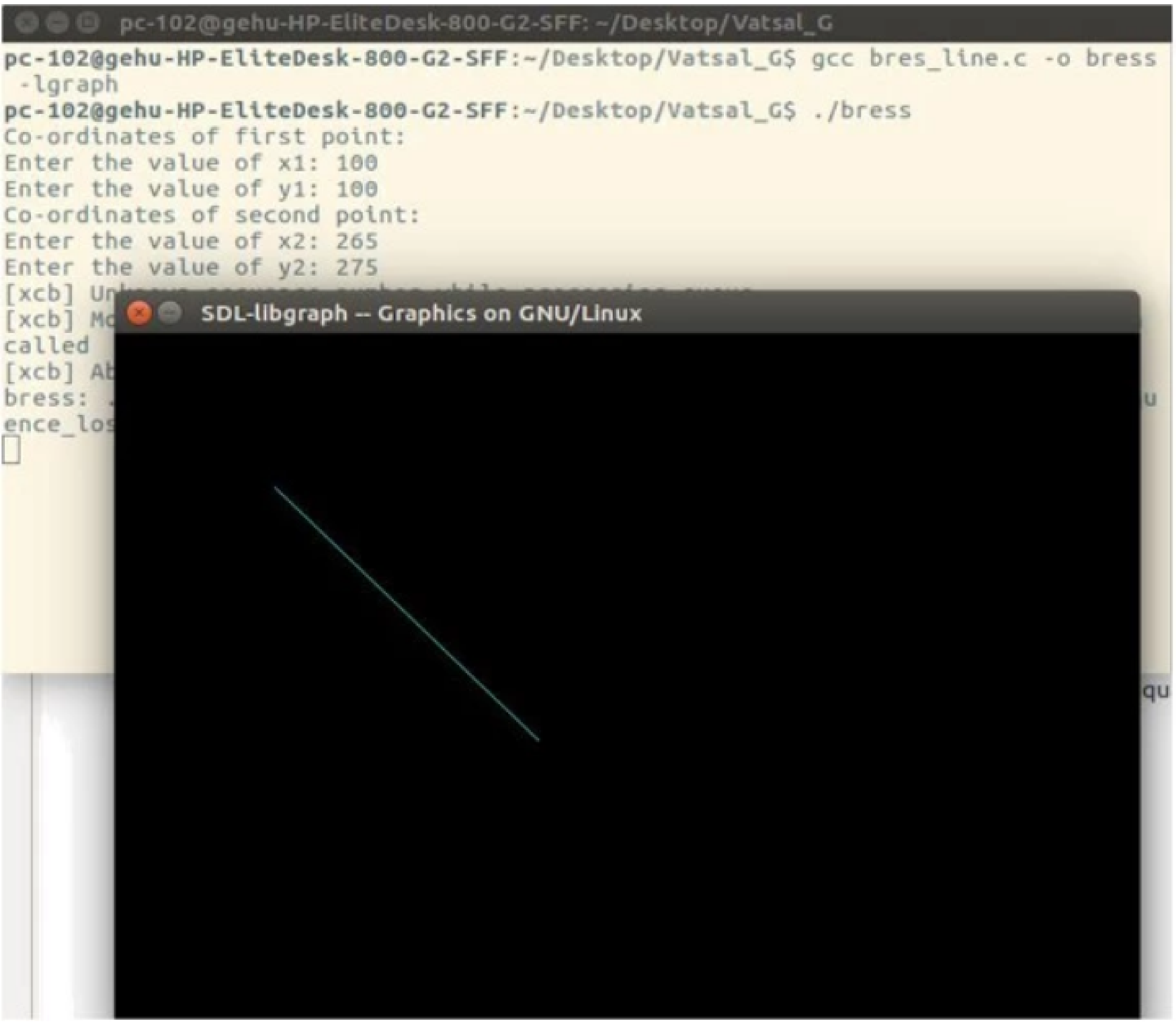
Step 6 :- Generate point at (x, y) coordinates

Step 7 :- check if whole line is generated.

```
if x >= xend
  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.
```



```
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
[xcb] Un
[xcb] Mo
called
[xcb] Ab
bress: .                                                                       u
ence_los
```

Name :- Himanshu Gusain　　　　Course :- BCA 'Sc'

Roll no. :- 05　　　　Subject :- Computer Graphics

Que 2 :- • Mid point circle generation algorithm :-

```
#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-x, y0-y, 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;
        }
      }
   }
```

Scanned By ScanIt

```
int main()
{
int gdriver = DETECT, gmode, error, x, y, r;
printf("Enter radius of circle:");
scanf("%d", &r);
printf("Enter co-ordinate 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:-

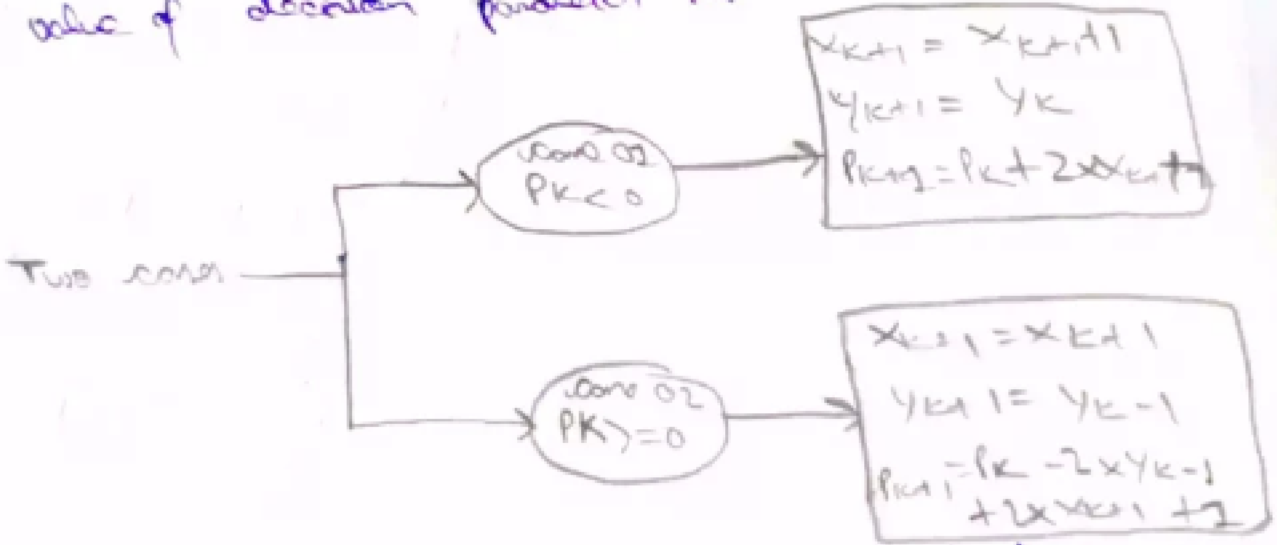
- $x_0 = 0$
- $y_0 = R$

Step 2:- Calculate the value of initial decision 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



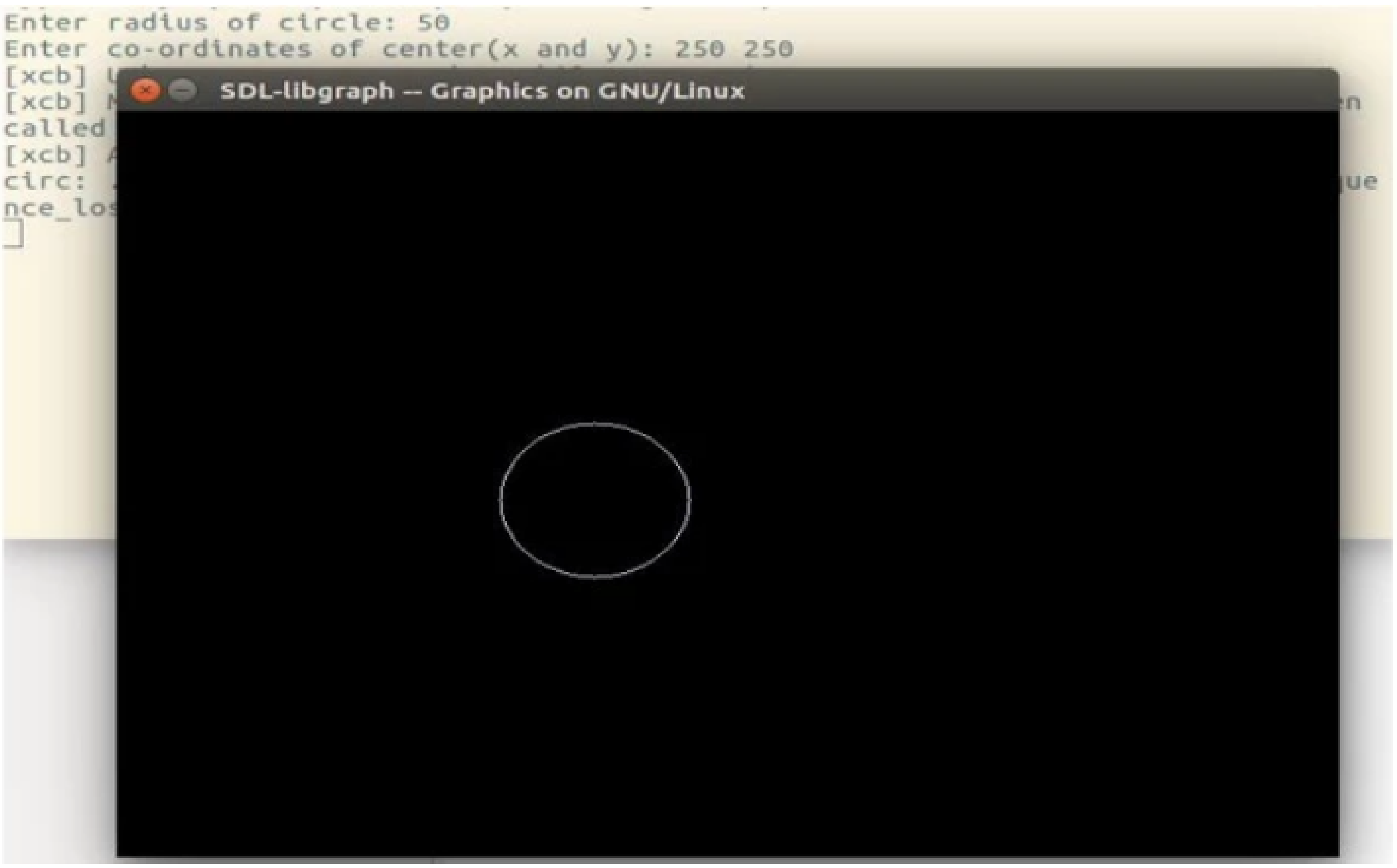
Step 4:- If the given center point (x_0, y_0) is not $(0,0)$, then do the following and plot the point

- $x_{plot} = x_c + x_0$
- $y_{plot} = y_c + y_0$

Here, (x_c, y_c) denotes the current value of x and y coordinates

Step 5:- Keep repeating step 03 and step 04 until $x_{plot} \geq y_{plot}$.

Step 6:- Generates all the points for one octant to find the points for other seven octants, follow the eight symmetry property of circle.



```
Enter radius of circle: 50
Enter co-ordinates of center(x and y): 250 250
[xcb] U
[xcb] M                                                                 en
called
[xcb] A
circ: .                                                                  ue
nce_los
```