NAME- VINIT KANYAL

ROLL NO- 1121168

SEM- 6 COURSE = BCA

PAPER NAME- COMPUTER GRAPHICS

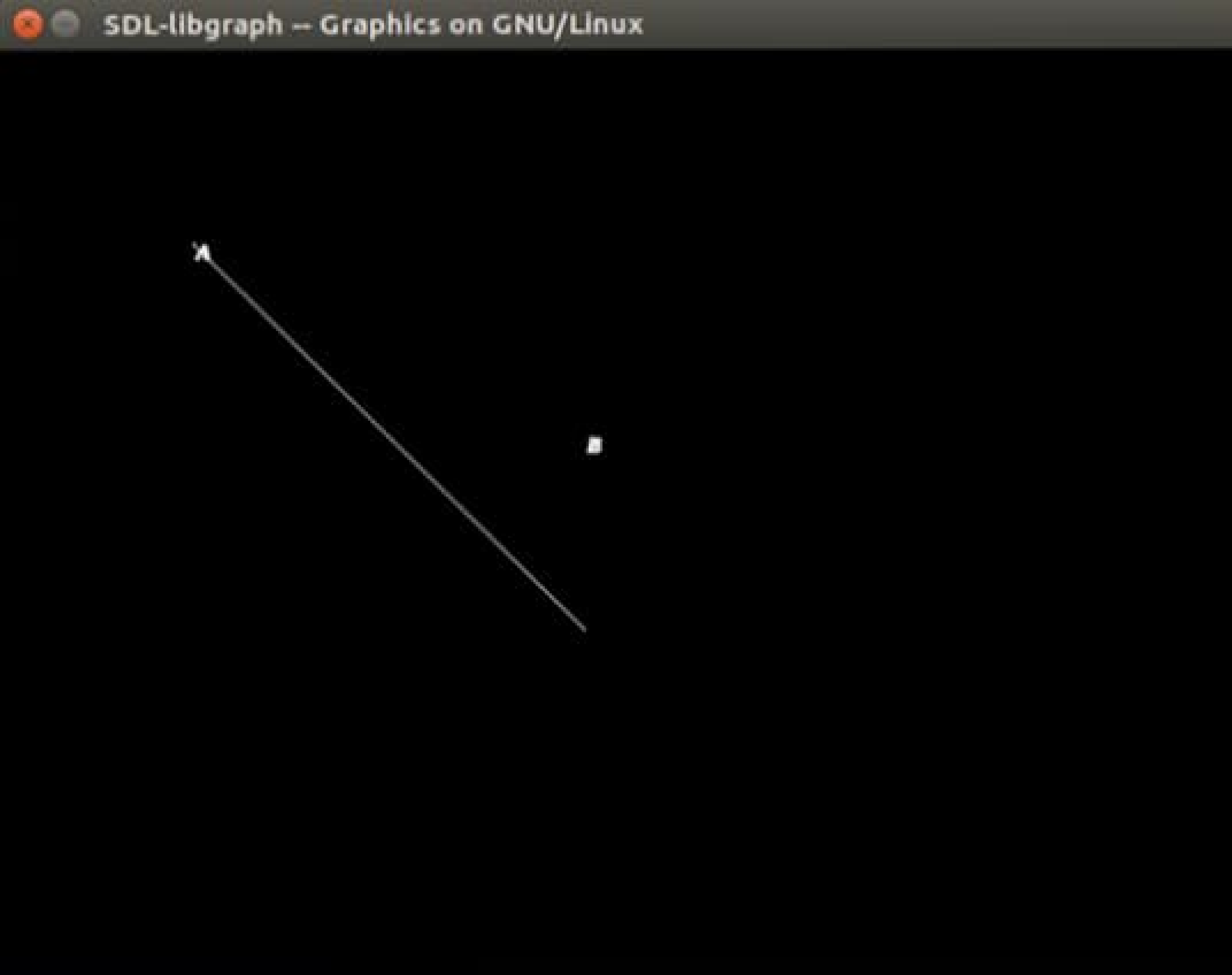
PAPER CODE - PBC-602

P1.)

```
# include <stdio.h>
# include <graphic.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");
outpixel(x1, y1, WHITE);
x=x1, y=y1;
while (step>0)
{ if (pk<0)
  {
   pkk = pk+2*dy;
  }
  else
  {
   pkk = pk+2*dy-2*dx;
   y++;
  }
  putpixel (rou(x), rou(y), WHITE),
  x++;
  step--;
}
getch() ;
return 0;
}
```

Algorithm:-

```
Step1 = Start
Step2 = Declare variable x1, x2, y1, y2, dx, dy in int
        and pk, pkk, x, y, step in float.
Step3 - Enter coordinates of x1, x2, y1, y2;
Step4 - Calculate dx = x2 - x1;
                  dy = y2 - y1;
                  pk = 2*dx - dy;

Step 5.) Initialize p = pk, x = x1, y = y1

Step 6.) Repeat step 7 to step 9 while i<=steps:

Step7) Check if pk<0 then
         putpixel (x, y, BLUE)
         x = x+1
         y = y;
         pkk = pk+(2*dy);

       Otherwise step 8:
Step 8.) putpixel (x, y, BLUE)

         x = x+1
         y = y+1
         pkk = pk+(2*dy) - (2*dx);

Step 9.) Increment i by one.

Step 10) Stop
```


SDL-libgraph -- Graphics on GNU/Linux

P2)

```
#include <graphic.h>
# include <stdio.h>
void midpoint (int midx, int midy, int r)
{
 int x=0, y=r, gd=0, gm, di, dnext;
   initgraph (&gd, &gm, "");
   di=1.2r-r;
  while (x<=y)
 {
   if (di >=0)
     {
       dnext  = di+2*(x-y)+1;
       x++;
       y--;
     }
    else
     {
       dnext = di+2*x+1;
       x++;
     }
   putpixel (x+midx, y+midy, 5);
   putpixel (y+midx, x+midy, 5);
   ~~putpixel~~ (y
    putpixel (-x+ midx, -y+midy, 5);
    putpixel (-y+ midx, -x +midy, 5);
    putpixel (-y+midx, x + midy, 5);
    putpixel (y+ midx, -x + midy, 5);
    putpixel (x +midx, -y+ midy, 5);
    putpixel (-x +midx, y+midy, 5);
    di = dnext;
 }
 getch();
 closegraph();
}
int main()
 {
   int gd=0, gm;
   int midx=0, midy=0, r=0;
   printf("Enter the coordinates (x,y):");
   scanf ("%d %d", &midx, &midy);
   printf("Enter the radius:");
   scanf ("%d", &r);
   midpoint (midx, midy, r)
 return 0;
}
```

Algorithm.

Step1 - Start

Step 2 - Plot the center coordinates (p_0, q_0) follows.

$p_0 = 0$, $q_0 = r$

Step 3 - Now, calculate the init decision paramet

$d_0 = 1-r$;

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 the first octant according to de.

Step 5 - follows these 2 case

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 $d_k >= 0$, then

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

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

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

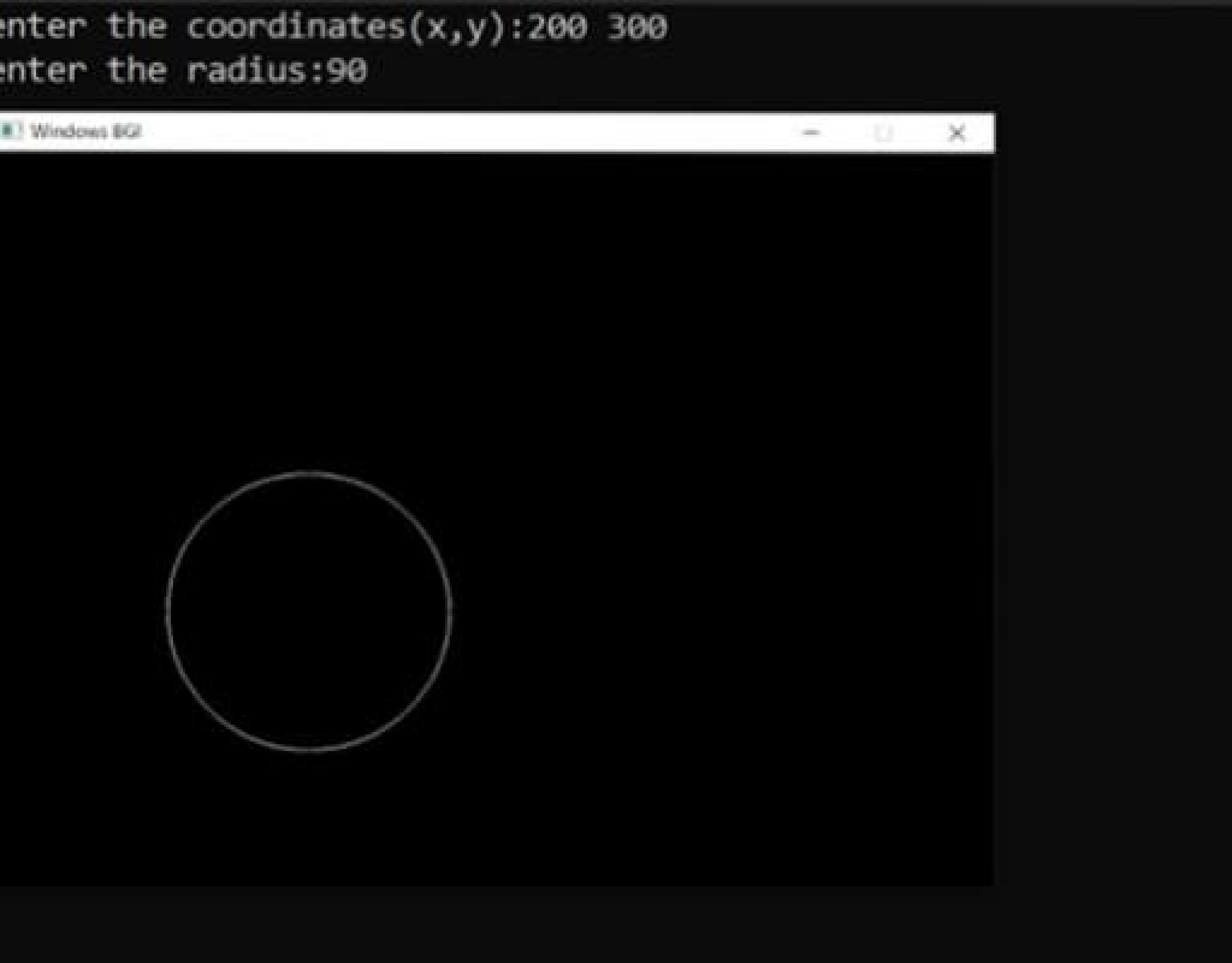
Step 6 - if center not (0,0) points will be

x coordinates $= x_c + p_0$

y coordinates $= y_c + q_0$

step 7 - Repeat Step 5 & 6 until $x >= y$

Step 8 - Stop.



```
enter the coordinates(x,y):200 300
enter the radius:90
```