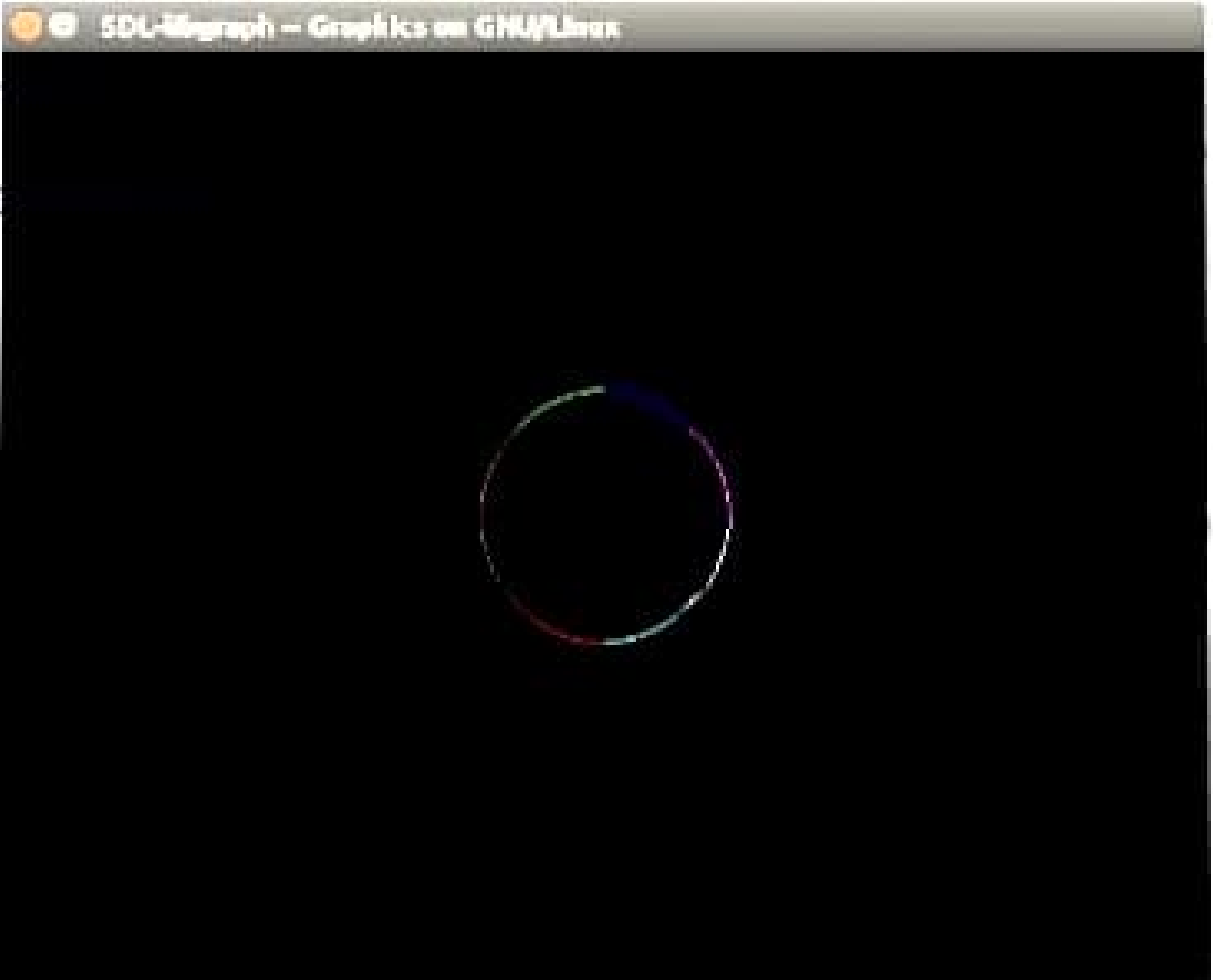
Saksham Chadha

BCA 6B

7121118

Program To Draw a circle using Bresenhams Circle Algorithm.

```
#include <stdio.h>
# include <graphics.h>.
 void main()
{
  int gd = DETECT, gm;
  int r, x, y, p, xc =320, yc=240;
 printf("Enter the Radius");
 scanf("%d", &r);
initgraph (&gd, &gm, "");
x=0;
y=r;
putpixel (xc+x, yc-y, 1);
p= 3-(2*r);
for(x=0; x<=y ; x++).
{
   if (p<0)
  {
    y=y;
  p= (p+(4*x)+6);
 }
else
{
 y=y-1;
p= p+((4*(x-y)+10));
 }
  putpixel (xc+x, yc-y, 1);
  putpixel (xc-x, yc-y, 2);
  putpixel ( xc+x, yc+y, 3);
  putpixel ( xc-x, yc+y, 4);
 putpixel (xc+y, yc-x, 5);
 putpixel (xc-y, yc+x, 6);
 putpixel ( xc+y, yc+x, 7);
 putpixel ( xc-y, yc+x, 8);
}
 getch ();
 closegraph ();
}.
```


SDL-libgraph – Graphics on GNU/Linux

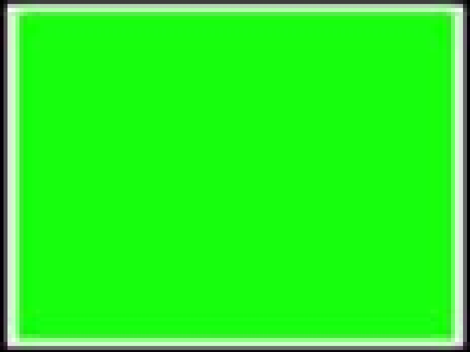
Algorithm

```
Step 1: Start
Step 2: Declare r, x, ~~x~~ y, p and initialize xc=320 and yc=240.
Step 3: Enter the value of Radius r.
Step 4: initialize x to 0 and y to r
Step 5: Calulate p=3-2r.
Step 6: Check next pixel to be scanned.
        if p<0
        then y=y and
        p= p+4x+6
    else
        increment y and
        p= p+ (4*(x-y)+10);
Step 7: Go to step 6 until x ~~b~~ is <=y.
Step 8: Plot points using concept of 8 way symmetry.
        Centre is at (xc, y).
Step 9: Go to step 8
Step 10: Stop.
```

```
# include < stdio.h>
# include <graphics.h>
# include<dos.h>
# include < conio.h>
void floodfill (intx, inty, int old, intnewcol).
 {
     int current;
     current= getpixel (x,y);
     if ( current == old).
     {
       delay (5);
    putpixel (x,y, newcol);
   floodfill (x+1, y, old, newcol);
   floodfill ( x-1,y, old, newcol);
   floodfill ( x, y+1, old , newcol);
   floodfill ( x, y-1, old, newcol);
   floodfill (x+1, y+1, old, newcol);
   floodfill (x-1, y+1, old, newcol);
   floodfill (x+1, y-1, old, newcol);
   floodfill (x-1, y-1, old, newcol);
   } }
```

```
    void main ()
   {
    int gd= DETECT, gm;
    initgraph (&gd ,&gm);
   Rectangle (50, 50, 150, 150);
   floodfill (70, 70, 0, 15);
   getch();
   closegraph ();

 }.
```



Floodfill Algorithm.

Step 1: Initialize the value of seed point (x,y), fcolor and dcol.

Step 2: Define boundary values of polygon

Step 3: Check if current seed point is of default color then repeat Step 4 and Step 5 till boundary pixels are reached.

if getpixel (x, y) = dcol then repeat 4 and 5.

Step 4: Change default color with fillcolor at seed point.

setpixel (x, y, fcol).

Step 5 → Recursively follow the procedure with 4 neighbourhood points.

```
floodfill (x-1, y, fcol, dcol)
floodfill (x+1, y, fcol, dcol)
floodfill (x, y-1, fcol, dcol)
floodfill (x, y+1, fcol, dcol)
floodfill (x+1, y+1, fcol, dcol)
floodfill (x-1, y+1, fcol, dcol)
floodfill (x+1, y-1, fcol, dcol)
floodfill (x-1, y-1, fcol, dcol)
```

Step 6: Exit