Set-B

Name :- Pooja Bisht

Course :- BCA - B (6th sem)

Roll NO :- 1121101 (20)

Paper Name :- Computer Graphics and Animation LAB.

Paper Type :- Regular (end-term Practical)

Email Id :- poojabisht031@gmail.com.

Source code :- (Program 1).

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

Pooja

```
        floodfill(x+1,y-1,old,new);
        floodfill(x-1,y-1,old,new);
    }
}
void main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    rectangle(50,50,150,150);
    floodfill(70,70,0,15);
    getch();
    closegraph();
}
```

Pooja

Algorithm :-

step 01 :- start.

step 02 :- Initialize the value of speed point (seedx, seedy), fcolor and dcol.

step 03 :- check if current seed point is of default color then repeat the st

step-03 :- Define the boundary values of the polygon.

step-04 :- check if the current seed point is of default color then repeat the steps 5 and 6 till the boundary pixels reached.

if getpixel(x,y) = dcol then repeat step 5 and 6.

step-05 :- change the default color with the fill color at the seed point.

setPixel (seedx, seedy, fcol)

Pooja

Step06 :- Recursively follow the procedure with four neighbourhood points.

floodfill (seedx - 1, seedy, fcol, dcol)
floodfill (seedx + 1, seedy, fcol, dcol)
floodfill (seedx, seedy - 1, fcol, dcol)
floodfill (seedx, seedy + 1, fcol, dcol)
floodfill (seedx - 1, seedy + 1, fcol, dcol)
floodfill (seedx + 1, seedy + 1, fcol, dcol)
floodfill (seedx + 1, seedy - 1, fcol, dcol)
floodfill (seedx - 1, seedy - 1, fcol, dcol)

Step07 :- Stop.

Pooja

SET-B

Program 3 :-

P3 :- Write An Algorithm and program to implement Breseham circle Drawing algorithm.

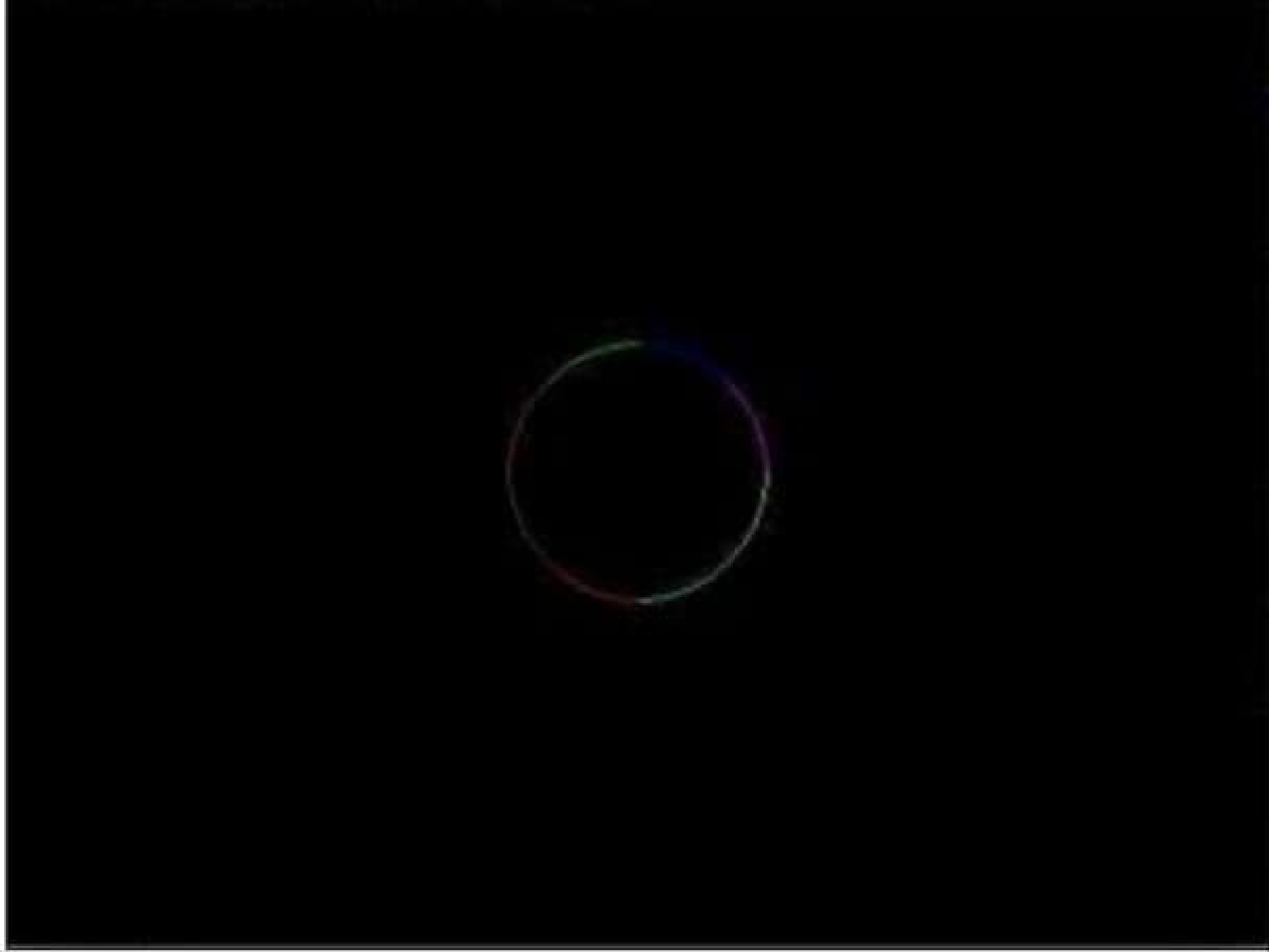
Source code :-

```
#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)
        {
```

Pooja

```
            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+x, yc-y, 5);
  putpixel (xc+x, yc-y, 6);
  putpixel (xc+x, yc-y, 7);
  putpixel (xc+x, yc-y, 8);
 }
getch();
closegraph();
}
```

Tooja



Bresenham's Circle Algorithm

Step 01 :- Start

Step 02 :- Declare p, q, x, y, r, d variables
p, q are coordinates of the center of the circle
r is the radius of the circle.

Step 03 :- Enter the value of r.

Step 04 :- Calculate $d = 3 - 2r$

Step 05 :- Initialize $x = 0$

Step 06 :- Check if the whole circle is scan converted
if $x >= y$
stop

Step 07 :- Plot eight points by using concepts of eight-way symmetry. The center is at (p, q). Current active pixel is (x, y).
putpixel(x+p, y+q)

Pooja

putpixel (y+p, x+q)
putpixel (-y+p, x+q)
putpixel (-x+p, y+q)
putpixel (-x+p, -y+q)
putpixel (-y+p, -x+q)
putpixel (y+p, -x+q)
putpixel (x+p, -y-q)

step 08 :- find location of next pixels to be scanned
if $d < 0$
then $d = d + 4x + 6$
increment $x = x + 1$
if $d \geq 0$
then $d = d + 4(x-y) + 10$
increment $x = x + 1$
decrement $y = y - 1$

step 09 :- Go to step 06.

step 10 :- Stop.

Yogi