Name : Muskaan Khetarpal

Father's Name : Mr. S.S. Khetarpal

Course : BCA (VI) - Sec B

University Roll No. 11 21086

Class Roll No. 06

Subject : Computer Graphics and Animation
(CTBC-602)

Muskaan

Q.3) Write an algorithm and program to implement Bresenham circle drawing Algorithm.

Algorithm

Step-1) Start Algorithm

Step-2) Declare p,q,x,y,r,d variables.

p,q are coordinates of the center of the circle

r is the radius of circle

Step-3) Enter the value of r

Step-4) Calculate d=3-2r

Step-5) Initialize x=0 and nbsy=r

Step-6) Check if the whole circle is scan converted

if x>=y

Stop

Step-7) Plot eight points by using concepts of eight way symmetry. The centre is at (p,q).

Current active pixel is (x,y)

putpixel (x+p, y+q)

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-8) 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-9) Go to step-6

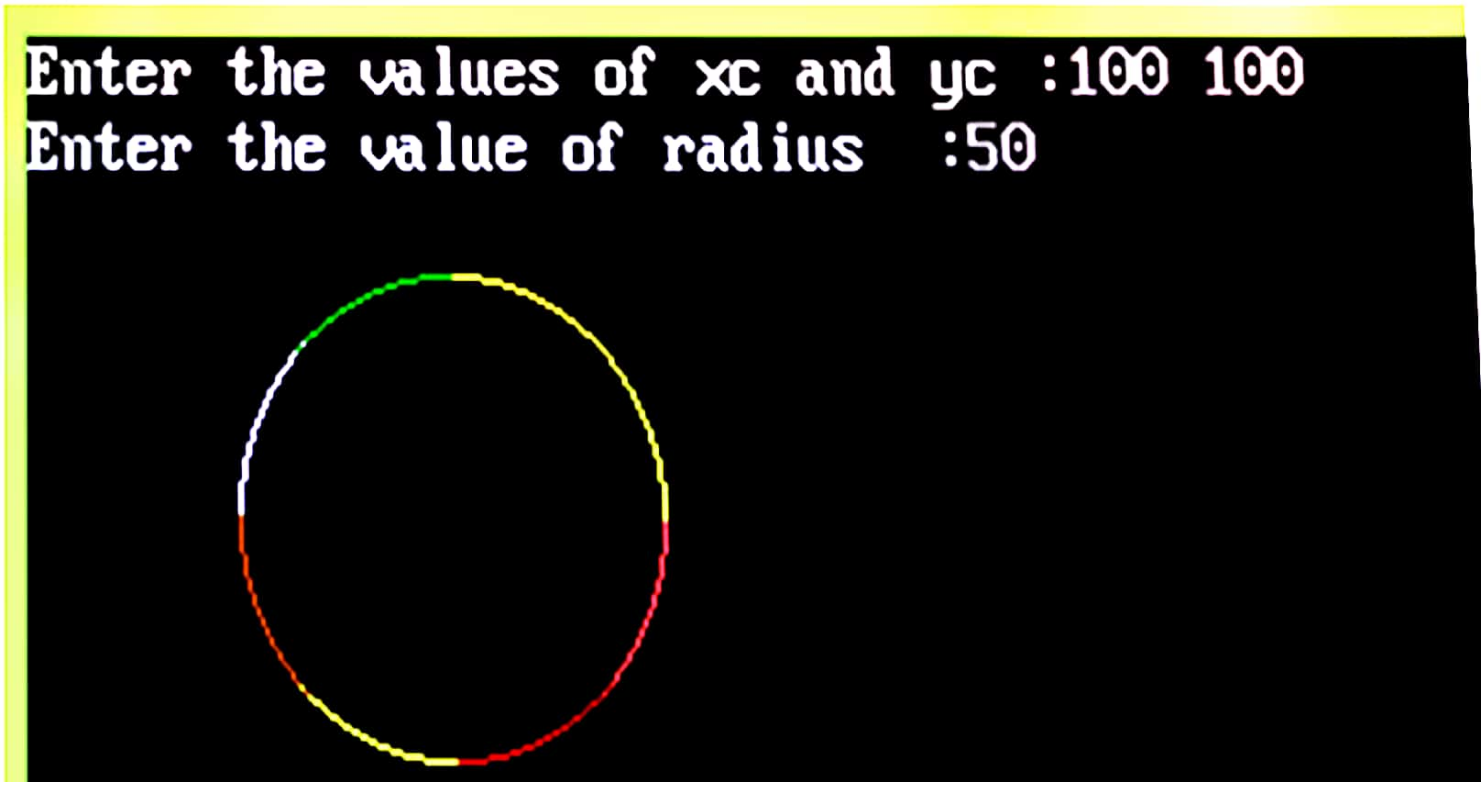
Step-10) Stop Algorithm

Code

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>

void EightWaySymmetricPlot(int xc, int yc, int x, int y)
{
  putpixel(x+xc, y+yc, RED);
  putpixel(x+xc, -y+yc, YELLOW);
  putpixel(-x+xc, -y+yc, GREEN);
  putpixel(-x+xc, y+yc, YELLOW);
  putpixel(y+xc, x+yc, 12);
  putpixel(y+xc, -x+yc, 14);
  putpixel(-y+xc, -x+yc, 15);
  putpixel(-y+xc, x+yc, 16);
}

void ~~Bresha~~ BresenhamCircle(int xc, int yc, int r)
{
  int x=0, y=r, d=3-(2*r);
  EightWaySymmetricPlot(xc, yc, x, y);
  while(xc=y)
  {
```

```
    if (d<=0)
    {
     d= d+(4*x)+6;
    }
    else
    {
     d=d+(4*x)-(4*y)+10;
      y=y-1;
     }
     x=x+1;
    EightWaySymmetricPlot(xc,yc,x,y);
   }
}

  int main(void)
 {
  int xc, yc, r, gd=DETECT, gm, errorcode;
  initgraph(&gd, &gm, "");
  errorcode = graphresult();
  if (errorcode != grOK)
  {
   printf("Graphics error: %s\n", grapherrormsg
                                    (errorcode));
   printf("Press any key to halt");
```

```
getch();
exit(1);
}
printf("Enter the values of xc and yc:");
scanf("%d%d",&xc,&yc);
printf("Enter the value of radius:");
scanf("%d",&r);
BresenhamCircle(xc,yc,r);
getch();
closegraph();
return 0;
}
```



9.1) Write an algorithm and program to implement floodfill algorithm using 8 connected method.

```
#include <stdio.h>
#include<graphics.h>
#include<conio.h>
void floodfill (int x, int y, int old, int newcol)
{
  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();
}
```

Algorithm for floodfill (8-connected method)

Step-1) Procedure floodfill (x, y, oldcolor, newcolor)

Step-2) If x or y is outside the screen, then return

Step-3) If color of getpixel (x, y) is same as old color, then

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)

Step-4) After filling all the colors

Step-5) Stop

Output: