**PROGRAM 13**

**Write a program to implement Boundary Fill algorithm.**

#include<stdio.h>

#include<graphics.h>

void boundaryFill(int x, int y, int f, int b)

{

if((getpixel(x,y)!=b)&&(getpixel(x,y)!=f))

{

putpixel(x,y,f);

boundaryFill(x+1,y,f,b);

boundaryFill(x-1,y,f,b);

boundaryFill(x,y+1,f,b);

boundaryFill(x,y-1,f,b);

}

}

void setpixel(int xc, int yc, int x, int y)

{

putpixel(xc+x,yc+y,15);

putpixel(xc+x,yc-y,15);

putpixel(xc-x,yc+y,15);

putpixel(xc-x,yc-y,15);

putpixel(xc+y,yc+x,15);

putpixel(xc+y,yc-x,15);

putpixel(xc-y,yc+x,15);

putpixel(xc-y,yc-x,15);

}

void midptcircle(int xc, int yc, int r)

{

int p = 1 - r;

int x= 0, y= r;

setpixel(xc,yc,x,y);

while(x<y)

{

x++;

if(p<0)

{

p+= 2\*x +1;

}

else

{

y--;

p+= 2\*(x-y) +1;

}

setpixel(xc,yc,x,y);

}

}

int main()

{

int xc, yc, r;

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "..\\");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

printf("Enter center of circle\n");

scanf("%d %d", &xc, &yc);

printf("Enter radius\n");

scanf("%d", &r);

midptcircle(xc,yc,r);

boundaryFill(xc,yc,15,15);

getch();

closegraph();

return 0;

}

**OUTPUT 13**



