

Name: Rahul Singh Shahi

Course: BCA

Sec: B

Roll No: 1121105 (24)

Subject: Computer Graphics

Ans 1:

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

```
void flood fill(int x, int y, int old, int newcol)
```

```
{
```

```
    int current;
```

```
    current = getpixel(x, y);
```

```
    if (current == old)
```

```
    {
```

```
        delay(5);
```

```
        putpixel(x, y, newcol);
```

```
        flood fill(x+1, y, old, newcol);
```

```
        flood fill(x-1, y, old, newcol);
```

```
        flood fill(x, y+1, old, newcol);
```

```
        flood fill(x, y-1, old, newcol);
```

```
        flood fill(x+1, y+1, old, newcol);
```

```
        flood fill(x-1, y+1, old, newcol);
```

```
        flood fill(x-1, y-1, old, newcol);
```

```
        flood fill(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:

Step 1: Start

Step 2: Draw the Rectangle using rectangle function

Step 3: Implement a connected flood fill with the

Co-ordinates x and y,

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);

Step 4: Stop



Bresenham's Circle Algorithm:

Name: Rahul Singh
Shah

Course: BCA

Roll: 1121105 (24)

Section: B

Step 1: Start Algorithm

Step 2: Declare P, q, x, y, r as Variables

P, q are coordinates of the center of the circle
 r is the Radius of the circle.

Step 3: Enter the value of r

Step 4: Calculate $d = 3 - 2r$

Step 5: Initialize $x = 0$

~~$y = 0$~~ $y = r$

Step 6: Check if the whole circle is scan converted

if $x \geq y$

Stop

Step 7: Plot eight points by using concepts of eight way symmetry. The center is at (P, q) current active pixel is (x, y)

Putpixel $(x+P, y+q)$

Putpixel $(y+P, x+q)$

Putpixel $(-y+P, y+q)$

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.

Program # include <graphics.h>

include <stdio.h>

include <stdlib.h>

include <conio.h>

include <math.h>

Void Eightway symmetric plot (int xc, int yc, int x, int

{
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, 12);

Putpixel(y+xc, -x+yc, 14);

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

Putpixel(-y+xc, x+yc, 6);

}

```
void Bresenham Circle (int xc, int yc, int r)
{
```

```
int x=0, y=r, d=3*(2*r);
```

```
Eight Way Symmetric plot(xc, yc, x, y);
```

```
while (x <= y)
```

```
{
  if (d <= 0)
```

```
    {
      d = d + (4*x) + 6;
```

```
    }
```

```
  else
```

```
    {
      d = d + (4*x) - (4*y) + 10;
      y = y - 1;
```

```
    }
```

```
    x = x + 1
```

```
Eight Way Symmetric plot(xc, yc, x, y);
```

```
}
```

```
}
```

```
int main (void)
```

```
{
```

```
int xc, yc, int driver = DETECT, gmode, errorCode;
```

```
int i; graph(&driver, &gmode);
```

```
errorCode = graphresult();
```

```
if (errorCode != 0)
```

```
{
```

```
printf("graph error: %s\n", grapherrmsg(error));
```

```
printf("press any key to halt");
```

```
getch();
```



```
exit(1);
```

```
}
```

```
Print ("Enter the value of xc and yc:");
```

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

```
Print ("Enter the value of Radius:");
```

```
scanf ("%d", &r);
```

```
Bresenham Circle (xc, yc, r);
```

```
getch();
```

```
close graph();
```

```
return 0;
```

```
}
```

Enter the values of xc and yc :100 100
Enter the value of radius :50

