

Name:- Manjeet Singh Negi

Roll no:- 1121081(02)

Course:- BCA-B-VI

Subject:- Computer Graphics

Ans 2.

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

int x1, y1, x2, y2, x3, y3, mx, my;

void draw();
void tri();
void main()
{
    int gd = DETECT, gm;
    int c;
    initgraph(&gd, &gm, "...\\bgi");
    printf("Enter the 1st point for the triangle:");
    scanf("%d%d", &x1, &y1);
    printf("Enter the 2nd point for the triangle:");
```

Manjeet

```
scanf ("%d%d", &x2, &y2);  
printf ("Enter the 3rd point for the  
triangle");  
scanf ("%d%d", &x3, &y3);
```

```
cleardevice();
```

```
draw();
```

```
getch();
```

```
tn();
```

```
getch();
```

```
}
```

```
void draw()
```

```
{
```

```
line(x1, y1, x2, y2);
```

```
line(x2, y2, x3, y3);
```

```
line(x3, y3, x1, y1);
```

```
}
```

```
void tn()
```

```
{
```

```
int x, y, a1, a2, a3, b1, b2, b3;
```

```
printf("Enter the transaction coordinates");
```

```
scanf ("%d%d", &x, &y);
```

```
cleardevice();
```

```
a1 = x1 + x;
```

```
b1 = y1 + y;
```

```
a2 = x2 + x;
```

```
b2 = y2 + y;
```

```
a3 = x3 + x;
```

```
b3 = y3 + y;
```

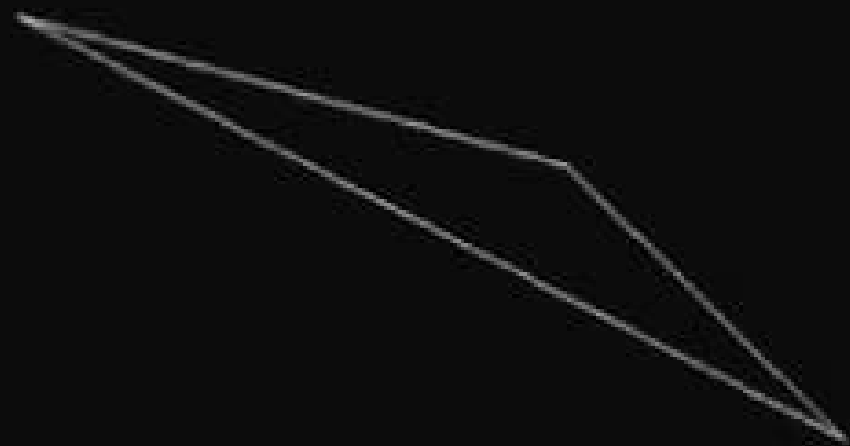
Majeed

line(a1, b1, a2, b2);
line(a2, b2, a3, b3);
line(a3, b3, a1, b1);

}

Manjeet

Enter the 1st point for the triangle:100 150
Enter the 2nd point for the triangle:320 210
Enter the 3rd point for the triangle:432 320





Enter the Transaction coordinates: 100 130



Names- Mayjeet Singh Negi
Courses- BCA-B-VI
Subjects- Computer graphics
Roll no:- 1121081(02)

Ans.

Bresenham's Circle 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 the circle

step 3: Enter the value

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

step 5: Initialize $x = 0$
& $nbxy = r$

step 6: check if the whole circle is
scan converted
if $x \geq y$
stop

step 7: plot eight point by using concepts
of eight-way symmetry. The center
is at (P, q) . current active pixel
is (x, y) .

Mayjeet

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 pixel to be scanned.

~~If~~ If $d < 0$

then $d = d + 4x + 6$

increment $x = x + 1$

~~If~~ If $d \geq 0$

then $d = d - 4x + 4y + 10$

increment $x = x + 1$

decrement $y = y - 1$

Step 9: Go to step 6

Step 10: stop Algorithm

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

```
void drawCircle (int xc, int yc, int x, int y)
```

```
{
```

```
    putpixel (xc+x, yc+y, RED)
    putpixel (xc-x, yc+y, RED)
    putpixel (xc+x, yc-y, RED)
    putpixel (xc-x, yc-y, RED)
    putpixel (xc+y, yc+x, RED)
    putpixel (xc-y, yc+x, RED)
    putpixel (xc+y, yc-x, RED)
    putpixel (xc-y, yc-x, RED)
```

```
}
```

```
void circleBres (int xc, int yc, int r)
```

```
{
```

```
    int x=0, y=r;
    int d = 3 - 2 * r;
    drawCircle (xc, yc, x, y);
    while (y >= x)
    {
        x++;
        if (d > 0)
        {
            y--;
            d = d + 4 * (x - y) + 10;
        }
        else
            d = d + 4 * x + 10;
        drawCircle (xc, yc, x, y);
        delay (50);
    }
```

```
}
```

Majeed

}

int main()

{

int xc=50, yc=50, r2=30;

int ga=DETECT, gm;

initgraph(&ga, &gm, "");

circlebres(xc, yc, r);

return 0;

}

Mandeep

