

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

```
void main()
```

```
{
```

```
    int gm;
```

```
    int gd = DETECT;
```

```
    int x1, y1, x2, y2, x3, y3, nx1, nx2, nx3, ny1, ny2, ny3, c;
```

```
    int sx, sy, xl, yl;
```

```
    float t;
```

```
    initgraph(&gd, &gm, "c:\\tc\\bgi");
```

```
    printf("Program for basic transactions");
```

```
    printf("Enter the points of triangle");
```

```
    setcolor(c);
```

```
    scanf("%d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3);
```

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

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

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

```
    getch();
```

```
    printf("1. Translation \n 2. Rotation \n 3. Scaling \n 4. Exit");
```

```
    printf("Enter your choice: ");
```

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

```
    switch(c)
```

case 1:

```
printf("\n Euler x-translation factor");  
scanf("%d %d", &x1, &y1);
```

$$nx1 = x1 + xt;$$

$$ny1 = y1 + yt;$$

$$nx2 = x2 + xt;$$

$$ny2 = y2 + yt;$$

$$nx3 = x3 + xt;$$

$$ny3 = y3 + yt;$$

```
line(nx1, ny1, nx2, ny2);
```

```
line(nx2, ny2, nx3, ny3);
```

```
line(nx3, ny3, nx1, ny1);
```

```
getch();
```

case 2:

```
printf("\n Euler the angle of rotation");
```

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

$$t = 3.14 * rt / 180;$$

$$nx1 = \text{abs}(x1 * \cos(t) - y1 * \sin(t));$$

$$ny1 = \text{abs}(x1 * \sin(t) + y1 * \cos(t));$$

$$nx2 = \text{abs}(x2 * \cos(t) - y2 * \sin(t));$$

$$ny2 = \text{abs}(x2 * \sin(t) + y2 * \cos(t));$$

$$nx3 = \text{abs}(x3 * \cos(t) - y3 * \sin(t));$$

$$ny3 = \text{abs}(x3 * \sin(t) + y3 * \cos(t));$$

nitish poudel

```

line (nx1,ny1,nx2,ny2);
line (nx2,ny2,nx3,ny3);
line (nx3,ny3,nx1,ny1);
getch();

```

Case 3:

```

printf("\nEnter the scaling factor");
scanf("%d %d", &SX, &SY);
nx1 = x1 * SX;
ny1 = y1 * SY;
nx2 = x2 * SX;
ny2 = y2 * SY;
nx3 = x3 * SX;
ny3 = y3 * SY;
line (nx1,ny1,nx2,ny2);
line (nx2,ny2,nx3,ny3);
line (nx3,ny3,nx1,ny1);
getch();

```

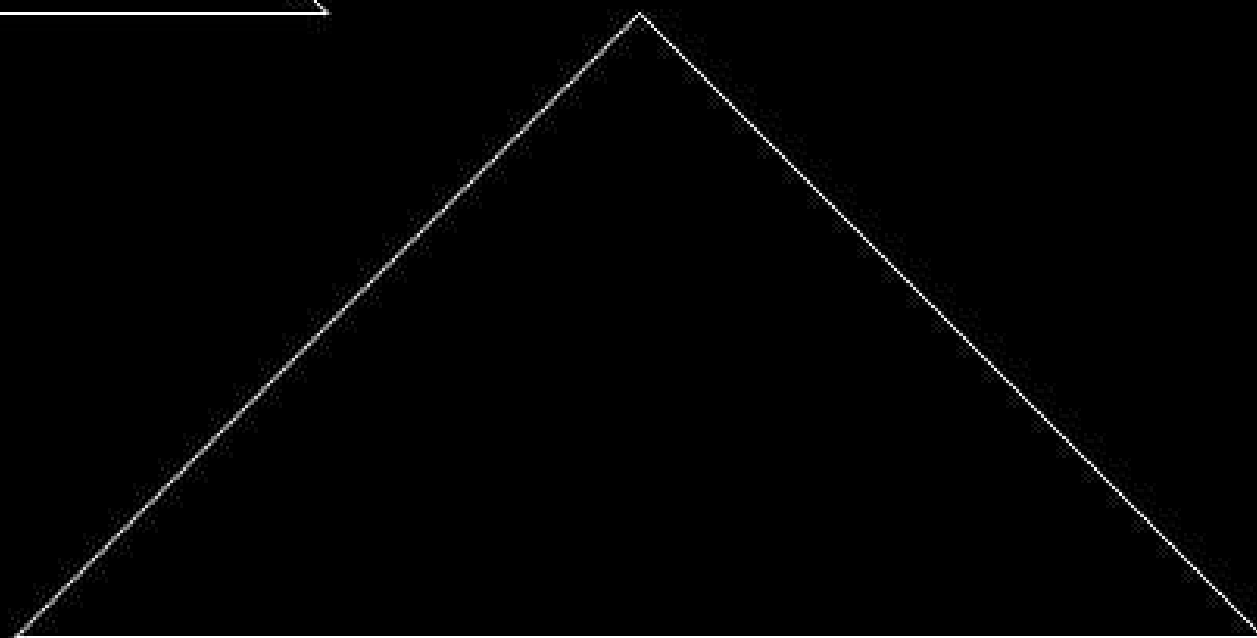
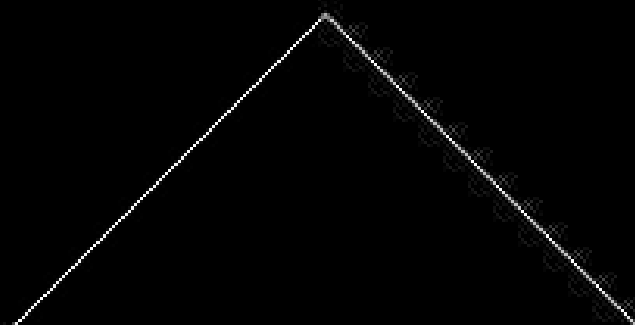
Case 4:

```

break;
default:
    printf("Enter valid choice.");
}
closegraph();
}

```

Waseem Raza



③ Algorithm:-

Step 1:- Start

Step 2:- Declare P, Q, X, Y, R , & 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 $x = 0$.

& $nbgy = R$

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 center 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 $(x + P, -y + Q)$

Wishy Rained

Step 8: Find location of bend 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

Witell Panch


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

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

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

```
{
```

```
{
```

```
int gd, gm, xc, yc, r, x, y, p;
```

```
detectgraph (&gd, &gm);
```

```
initgraph (&gd, &gm, "C:\\TurboC3\\3GI");
```

```
printf("Enter center of circle: ");
```

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

```
printf("Enter radius of circle: ");
```

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

```
x = 0;
```

```
y = r;
```

```
p = 3 - 2 * r;
```

```
pixel(xc, yc, x, y);
```

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

```
{ if (p < 0)
```

```
{ x++;
```

```
p = p + 4 * x + 6;
```

```
}
```

```
else
```

```
{ x++;
```

```
y--;
```

```
p = p + 4 * (x - y) + 10;
```

while (x < y)

```
pixel(xc, yc, h, y);
```

```
}
```

```
  getch();
```

```
  closegraph(c);
```

```
  return 0;
```

```
}
```

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

```
{
```

```
  putpixel(xc+x, yc+y, WHITE);
```

```
  putpixel(xc+x, yc-y, WHITE);
```

```
  putpixel(xc-x, yc+y, WHITE);
```

```
  putpixel(xc-x, yc-y, WHITE);
```

```
  putpixel(xc+y, yc+x, WHITE);
```

```
  putpixel(xc+y, yc-x, WHITE);
```

```
  putpixel(xc-y, yc+x, WHITE);
```

```
  putpixel(xc-y, yc-x, WHITE);
```

```
}
```


Enter the value of Xc 100
Enter the value of Yc 140
Enter the Radius of circle

97

