**Practical No: 03**

**Aim :** Write a c program to perform 2-D Rotation Transformation in Geometrical Transformation

**Performed By :** Vedant Bhojane

**Class:** BCA-III SEM-V

**Date:** 24/08/24

**Code :**

#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <math.h>

void DrawTriangle(int x1, int y1, int x2, int y2, int x3, int y3);

void RotateTriangle(int x1, int y1, int x2, int y2, int x3, int y3, float angle);

int main()

{

int gd = DETECT, gm;

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

float angle;

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

printf("Enter coordinate of a: ");

scanf("%d%d", &x1, &y1);

printf("Enter coordinate of b: ");

scanf("%d%d", &x2, &y2);

printf("Enter coordinate of c: ");

scanf("%d%d", &x3, &y3);

DrawTriangle(x1, y1, x2, y2, x3, y3);

printf("Enter the angle for rotation (in degrees): ");

scanf("%f", &angle);

RotateTriangle(x1, y1, x2, y2, x3, y3, angle);

getch();

closegraph();

return 0;

}

void DrawTriangle(int x1, int y1, int x2, int y2, int x3, int y3)

{

line(x1, y1, x2, y2);

line(x2, y2, x3, y3);

line(x3, y3, x1, y1);

}

void RotateTriangle(int x1, int y1, int x2, int y2, int x3, int y3, float angle)

{

int p = x2, q = y2;

float radianAngle = (angle \* 3.14) / 180.0;

int a1 = p + (x1 - p) \* cos(radianAngle) - (y1 - q) \* sin(radianAngle);

int b1 = q + (x1 - p) \* sin(radianAngle) + (y1 - q) \* cos(radianAngle);

int a2 = p + (x2 - p) \* cos(radianAngle) - (y2 - q) \* sin(radianAngle);

int b2 = q + (x2 - p) \* sin(radianAngle) + (y2 - q) \* cos(radianAngle);

int a3 = p + (x3 - p) \* cos(radianAngle) - (y3 - q) \* sin(radianAngle);

int b3 = q + (x3 - p) \* sin(radianAngle) + (y3 - q) \* cos(radianAngle);

setcolor(1);

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

}

**Output :**

