**EXERCISE - 3**

**AIM :** To implement 2D transformations such as translation, scaling, and rotation on a 2D object using C programming and a graphics library.

**Procedure (Using Bresenham’s Algorithm)**

2D transformations are used to manipulate objects in a two-dimensional plane. Common transformations include:

1. **Translation:** Moves an object from one location to another by adding offsets to the coordinates.

x′=x+tx,y′=y+ty

1. **Scaling:** Resizes an object by scaling factors Sx and Sy​.

x′=x⋅Sx, y′=y⋅Sy​

1. **Rotation:** Rotates an object around the origin by an angle θ

x′=x⋅cos⁡θ−y⋅sin⁡θ, y′=x⋅sin⁡θ+y⋅cos⁡θ

**Step 1 : Input the object coordinates:**  
Define the vertices of the object (e.g., a triangle or square).

**Step 2 :**  **Choose the transformation type:**  
Translation, scaling, or rotation.

**Step 3 :**  **Apply the transformation:**  
Use the appropriate formulas to calculate the transformed coordinates.

**Step 4 :**  **Display the results:**  
Render the original and transformed objects on the screen.

**SAMPLE CODE:**

##include <stdio.h>

#include <graphics.h>

#include <math.h>

*// Function to draw the object (triangle)*

void drawObject(int x[], int y[], int n, int color) {

int i;

setcolor(color);

    for (i=0; i<n; i++) {

    line(x[i], y[i], x[(i + 1) % n], y[(i + 1) % n]);

    }

}

*// Translation*

void translate(int x[], int y[], int n, int tx, int ty) {

    int i;

    for (i = 0; i < n; i++) {

    x[i] += tx;

    y[i] += ty;

    }

}

*// Scaling*

void scale(int x[], int y[], int n, float sx, float sy) {

    int i;

    for (i = 0; i < n; i++) {

    x[i] = (int)(x[i] \* sx);

    y[i] = (int)(y[i] \* sy);

    }

}

*// Rotation*

void rotate(int x[], int y[], int n, float angle) {

    int i;

    float rad = angle \* (M\_PI / 180.0); *// Convert to radians*

    for (i = 0; i < n; i++) {

    int tempX = x[i], tempY = y[i];

    x[i] = (int)(tempX \* cos(rad) - tempY \* sin(rad));

    y[i] = (int)(tempX \* sin(rad) + tempY \* cos(rad));

    }

}

int main()

{

    int gd = DETECT, gm;

    int x[] = {100, 200, 150}; *// Triangle vertices*

    int y[] = {100, 100, 50};

    int n = 3; *// Number of vertices*

    int tx=50, ty=30;

    float sx=1.5,sy=1.5;

    float angle=45;

    initgraph(&gd,&gm,"C:\\TC\\BGI");

*// Draw the original object*

    printf("\n\t\t\t2D Transformations");

    printf("\nOriginal Object:");

    setcolor(WHITE);

    drawObject(x, y, n, WHITE);

    delay(1000);

*// Perform Translation*

    printf("\n\n\n\n\nTranslation:");

    translate(x, y, n, tx, ty);

    setcolor(GREEN);

    drawObject(x, y, n, GREEN);

    delay(1000);

*// Perform Scaling*

    printf("\n\n\n\n\nScaling:");

    scale(x, y, n, sx, sy);

    setcolor(RED);

    drawObject(x, y, n, RED);

    delay(1000);

*// Perform Rotation*

    printf("\n\n\n\n\nRotation:");

    rotate(x, y, n, angle);

    setcolor(BLUE);

    drawObject(x, y, n, BLUE);

*// Wait for user input to close*

    getch();

    closegraph();

    return 0;

}

**OUTPUT:**

