**Practical No: 02**

**Aim :** Write a c graphics program to perform 2D Scaling Transformation in Geometrical

Transformation

**Performed By** : Aditya Undre

**Class :** BCA III Sem-V

**Date :**29/08/2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code :**

#include <stdio.h>

#include <graphics.h>

#include <math.h>

#include <conio.h>

int gd = DETECT, gm;

int n, x[100], y[100], i;

float sfx, sfy;

void draw();

void scale();

void main()

{

clrscr();

printf("Enter number of sides: ");

scanf("%d",&n);

printf("Enter coordinates x, y for each point: ");

for (i = 0; i < n; i++)

scanf("%d%d",&x[i],&y[i]);

printf("Enter scale factors (sfx, sfy): ");

scanf("%d%d",&sfx,&sfy);

initgraph(&gd, &gm, (char\*)"");

cleardevice();

setcolor(WHITE);

draw();

scale();

setcolor(YELLOW);

draw();

getch();

}

void draw()

{

for (i = 0; i < n; i++) {

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

}

}

void scale()

{

for (i = 0; i < n; i++) {

x[i] = x[0] + (int)((float)(x[i] - x[0]) \* sfx);

y[i] = y[0] + (int)((float)(y[i] - y[0]) \* sfx);

}

}

**Output:**

