

## Lab 5: 2D Transformations

### **LAB SHEET**

**Theory:** Write down the transformation matrix of 2D translation, rotation and scaling.

**Lab works:**

**Question 1: Write a code in C to implement 2D translation of a rectangle.**

```
#include <stdio.h>

#include <graphics.h>

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, NULL);
    int x1 = 200, y1 = 150, x2 = 300, y2 = 250;
    int tx = 50, ty = 50;
    printf("Rectangle before translation\n");
    rectangle(x1, y1, x2, y2);
    printf("Rectangle after translation\n");
    rectangle(x1 + tx, y1 + ty, x2 + tx, y2 + ty);
    getch();
    closegraph();
    return 0;
}
```

**Question 2: Write a code in C to implement rotation of a line.**

```
#include<graphics.h>

#include<stdio.h>

#include<conio.h>
```

```

#include<math.h>

int main()
{
    int gd=DETECT,gm;
    int pivot_x,pivot_y,x,y;
    double degree,radian;
    int rotated_point_x,rotated_point_y;
    initgraph(&gd,&gm, NULL);
    cleardevice();
    printf("\t\t ROTATION \n");
    printf("\n Enter an initial coordinates of the line = ");
    scanf("%d %d",&pivot_x,&pivot_y);
    printf("\n Enter a final coordinates of the line = ");
    scanf("%d %d",&x,&y);
    line(pivot_x,pivot_y,x,y);
    printf("\n\n Now, Enter a degree = ");
    scanf("%lf",&degree);
    radian=degree*0.01745;
    rotated_point_x=(int)(pivot_x +((x-pivot_x)*cos(radian)-(y-pivot_y)*sin(radian)));
    rotated_point_y=(int)(pivot_y +((x-pivot_x)*sin(radian)+(y-pivot_y)*cos(radian)));
    setcolor(RED);
    line(pivot_x,pivot_y,rotated_point_x,rotated_point_y);
    getch();
    closegraph();
    return 0;
}

```

```
}
```

**Question 3: Write a code in C to implement scaling of a triangle.**

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

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

```
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);
```

```
}
```

```
int main() {
```

```
    int gd = DETECT, gm;
```

```
    initgraph(&gd, &gm, NULL);
```

```
    int x1 = 200, y1 = 200, x2 = 300, y2 = 300, x3 = 250, y3 = 100;
```

```
    float sx, sy;
```

```
    printf("Enter the scaling factors (sx sy): ");
```

```
    scanf("%f %f", &sx, &sy);
```

```
    printf("Triangle before scaling\n");
```

```
    drawTriangle(x1, y1, x2, y2, x3, y3);
```

```
    int new_x1 = x1 * sx;
```

```
    int new_y1 = y1 * sy;
```

```
    int new_x2 = x2 * sx;
```

```
    int new_y2 = y2 * sy;
```

```
    int new_x3 = x3 * sx;
```

```
    int new_y3 = y3 * sy;
```

```
printf("Triangle after scaling\n");  
drawTriangle(new_x1, new_y1, new_x2, new_y2, new_x3, new_y3);  
getch();  
closegraph();  
return 0;  
}
```

## **Conclusion**

- What did u learn?