

Name - Laxmikant S Babaleshwar
Class - SE -AI&DS- C1
Roll No - 20

```
//2D Transformation
#include<iostream>
#include<graphics.h>
using namespace std;
int main()
{
    int gd=DETECT,gm,s;
    initgraph(&gd,&gm,NULL);
    cout<<"MENU\n1.Translation\n2.Rotation\n3.Scaling\n"<<endl;
    cout<<"Enter your choice:";
    cin>>s;
    switch(s)
    {
        case 1:
        {
            int x1=200,y1=150,x2=300,y2=250;
            int tx=50,ty=50;
            cout<<"Rectangle before translation"<<endl;
            setcolor(CYAN);
            rectangle(x1,y1,x2,y2);
            setcolor(GREEN);
            cout<<"Rectangle after translation"<<endl;
            rectangle(x1+tx,y1+ty,x2+tx,y2+ty);
            getch();
            break;
        }
        case 2:
        {
            int x1 = 200, y1 = 200; // Top-left corner (pivot point)
            int x2 = 300, y2 = 200; // Top-right corner
            int x3 = 300, y3 = 300; // Bottom-right corner
            int x4 = 200, y4 = 300; // Bottom-left corner
            double angle;
            cout << "Rectangle before rotation" << endl;
            setcolor(CYAN);
            rectangle(x1, y1, x3, y3);
            cout << "Angle of rotation (in degrees): ";
            cin >> angle;
            double radian = (angle * 3.14159265) / 180.0;
```

```

        auto rotatePoint = [&](int &x, int &y)
        {
            int x_new = x1 + (x - x1) * cos(radian) - (y - y1) * sin(radian);
            int y_new = y1 + (x - x1) * sin(radian) + (y - y1) * cos(radian);
            x = x_new;
            y = y_new;
        };

        rotatePoint(x2, y2);
        rotatePoint(x3, y3);
        rotatePoint(x4, y4);

        setcolor(GREEN);
        line(x1, y1, x2, y2);
        line(x2, y2, x3, y3);
        line(x3, y3, x4, y4);
        line(x4, y4, x1, y1);
        getch();
        break;
    }
    case 3:
    {
        int x1=30,y1=30,x2=70,y2=70,sy=2,sx=2;
        cout<<"Before scaling"<<endl;
        setcolor(CYAN);
        rectangle(x1,y1,x2,y2);
        cout<<"After scaling"<<endl;
        setcolor(GREEN);
        rectangle(x1*sx,y1*sy,x2*sx,y2*sy);
        getch();
        break;
    }
    default:
    {
        cout<<"Invalid Selection"<<endl;
        break;
    }

}
closegraph();
return 0;
}

```

COMMAND:

```
(base) oem@oem-OptiPlex-3090:~$ g++ transform4.cpp -o transform4 -lgraph
(base) oem@oem-OptiPlex-3090:~$ ./transform4
```

OUTPUT:

