Name:Amay Dilip Mapari
Roll No.:21154
Class:SE1
Batch:G1
Assignment No.:9

Code:

```
#include<iostream>

#include<math.h>

#include<GL/glut.h>

using namespace std;


typedef float Matrix4 [3][3];


Matrix4 theMatrix;

static GLfloat input[4][2]=

{

   {40,40},{90,40},{90,90},{40,90}


};


float output[4][2];

float tx,ty;

float sx,sy;

float angle;


int choice;
```

```
void setIdentityM(Matrix4 m)
{
for(int i=0;i<3;i++)
   for(int j=0;j<3;j++)
       m[i][j]=(i==j);
}

void translate(int tx,int ty)
{

for(int i=0;i<4;i++)
{
output[i][0]=input[i][0]+tx;
output[i][1]=input[i][1]+ty;
}
}
void scale(int sx,int sy)
{
   theMatrix[0][0]=sx;
   theMatrix[1][1]=sy;
}

void RotateZ(float angle) //parallel to z
{
```

```
 angle = angle*3.14/180;
 theMatrix[0][0] = cos(angle);
 theMatrix[0][1] = sin(angle);
 theMatrix[1][0] = -sin(angle);
 theMatrix[1][1] = cos(angle);

}

void multiplyM()
{
//We Don't require 4th row and column in scaling and rotation
//[8][3]=[8][3]*[3][3] //4th not used
for(int i=0;i<4;i++)
 {
   for(int j=0;j<2;j++)
   {
      output[i][j]=0;
      for(int k=0;k<2;k++)
      {
         output[i][j]=output[i][j]+input[i][k]*theMatrix[k][j];
      }
   }
}

}
```

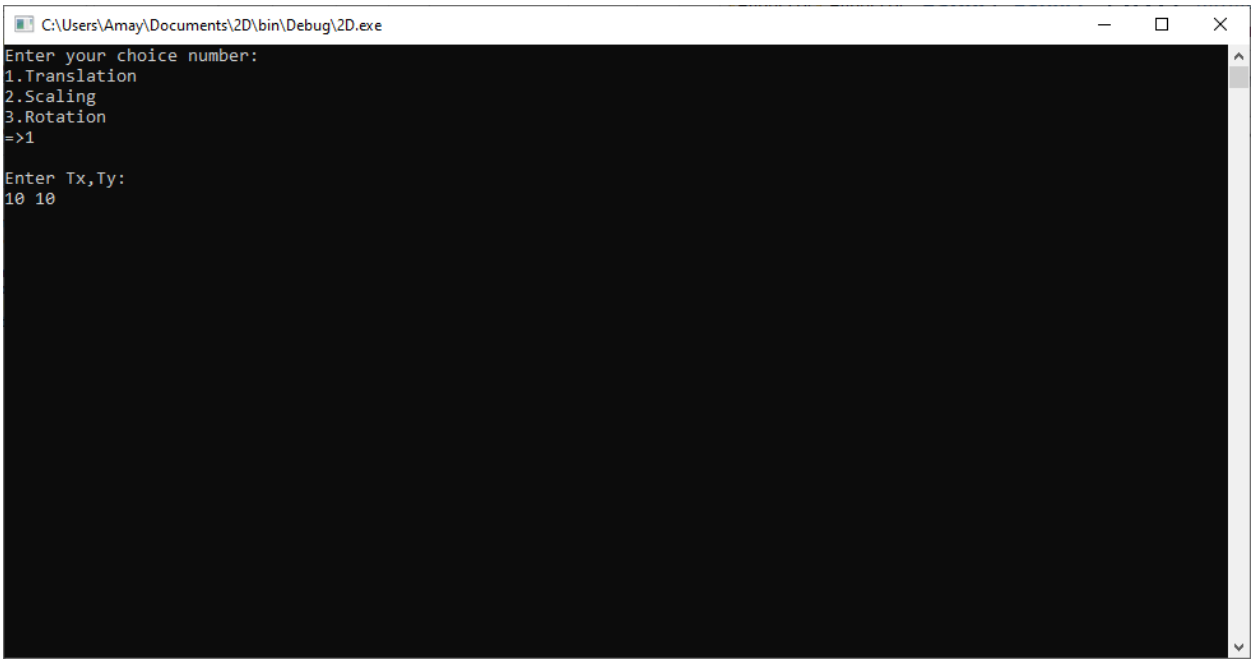
```
void draw(float a[4][2])
{



   glBegin(GL_QUADS);
    glColor3f(0.7,0.4,0.5); //behind
   glVertex3fv(a[0]);
   glVertex3fv(a[1]);
   glVertex3fv(a[2]);
   glVertex3fv(a[3]);
glEnd();
}


void init()
{
   glClearColor(1.0,1.0,1.0,1.0); //set backgrond color to white
   glOrtho(-454.0,454.0,-250.0,250.0,-250.0,250.0);
   // Set the no. of Co-ordinates along X & Y axes and their gappings
   glEnable(GL_DEPTH_TEST);
    // To Render the surfaces Properly according to their depths
}


void display()
{
```

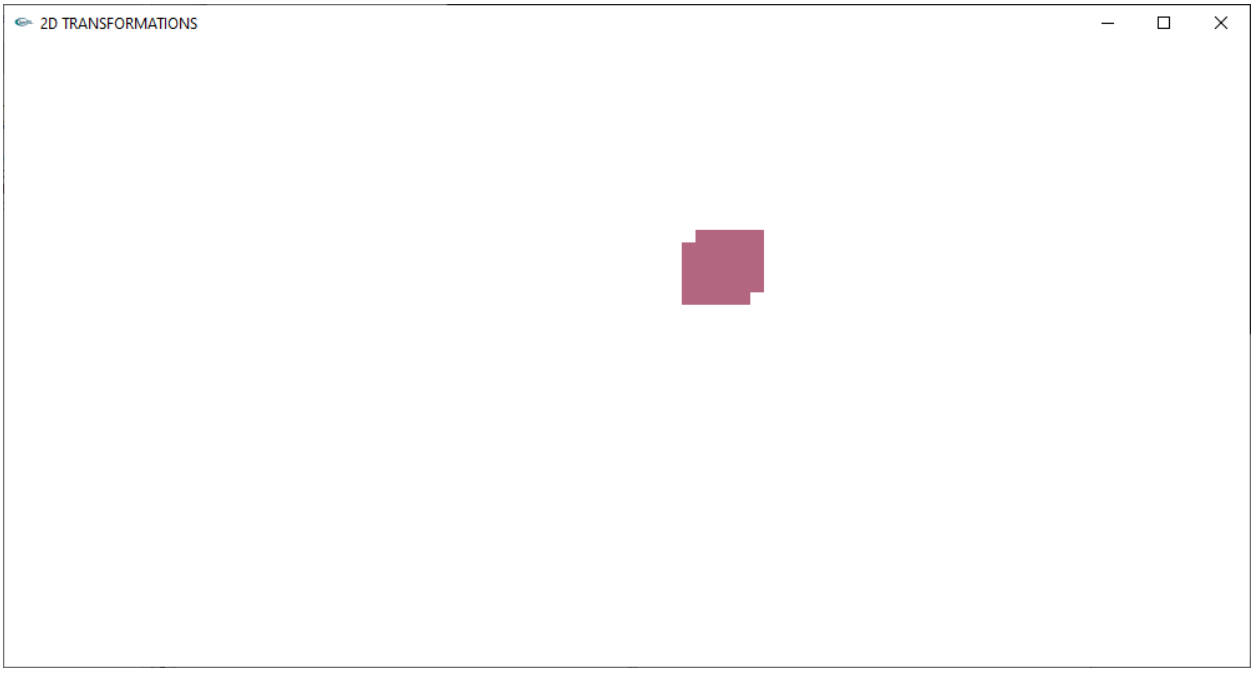
```
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);


glColor3f(1.0,0.0,0.0);
draw(input);
setIdentityM(theMatrix);
switch(choice)
{
case 1:
   translate(tx,ty);
   break;
 case 2:
   scale(sx,sy);
multiplyM();
   break;
 case 3:
      RotateZ(angle);
multiplyM();
   break;
}


draw(output);
glFlush();
}


int main(int argc, char** argv)
```

```
{
   glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
   glutInitWindowSize(1000,500);
   glutInitWindowPosition(0,0);
   glutCreateWindow("2D TRANSFORMATIONS");
   init();
   cout<<"Enter your choice number:\n1.Translation\n2.Scaling\n3.Rotation\n=>";
   cin>>choice;
   switch (choice) {

   case 1:
      cout<<"\nEnter Tx,Ty: \n";
      cin>>tx>>ty;
      break;
   case 2:
      cout<<"\nEnter Sx,Sy: \n";
      cin>>sx>>sy;
      break;
   case 3:

         cout<<"\nEnter Rotation angle: ";
         cin>>angle;
         break;
   default:
```

```
        break;
    }
    glutDisplayFunc(display);
    glutMainLoop();
return 0;
}
```

Output:




2D TRANSFORMATIONS