**Practical Question 6: 2D Transformation**

**write a program to apply various 2D transformation on a 2D object (use homogenous coordinates)**

#include<iostream>

#include<conio.h>

#include<graphics.h>

#include<math.h>

using namespace std;

struct point

{

float x,y,z;

};

class matrix

{

private:

point \*p;

point t[3];

point p1[1];

int points;

public:

int c = 400;

void axis()

{

setcolor(WHITE);

line(c,0,c,c\*2);

line(0,c,c\*2,c);

}

void plot()

{

for(int i=0;i<points;i++)

line(c+p[i].x, c-p[i].y, c+p[(i+1)%points].x, c-p[(i+1)%points].y);

}

void mat()

{

cout<<endl;

for(int i=0;i<points;i++)

cout<<p[i].x<<" "<<p[i].y<<" "<<p[i].z<<endl;

}

void read()

{

cout<<"Enter no. of points :"<<endl;

cin>>points;

p=new point[points];

cout<<"Note:Enter value between 0 and 400"<<endl;

cout<<"Enter coordinates:"<<endl;

for(int i=0;i<points;i++)

{

cout<<"Enter P"<<i+1<<" : "<<endl;

cin>>p[i].x>>p[i].y;

p[i].z=1;

}

setcolor(RED);

mat();

plot();

}

void trans\_mat()

{

int ch;

cout<<"1) Reflection"<<endl;

cout<<"2) Rotation"<<endl;

cout<<"3) Scaling"<<endl;

cout<<"4) Translation"<<endl;

cout<<"5) Shearing"<<endl;

cin>>ch;

switch(ch)

{

case 1:

{

cout<<"1) About x-axis"<<endl;

cout<<"2) About y-axis"<<endl;

cin>>ch;

switch(ch)

{

case 1:

{

t[0].x=1; t[0].y=0; t[0].z=0;

t[1].x=0; t[1].y=-1; t[1].z=0;

t[2].x=0; t[2].y=0; t[2].z=1;

break;

}

case 2:

{

t[0].x=-1; t[0].y=0; t[0].z=0;

t[1].x=0; t[1].y=1; t[1].z=0;

t[2].x=0; t[2].y=0; t[2].z=1;

break;

}

default:

{

cout<<"Wrong Choice Entered"<<endl;

break;

}

}

break;

}

case 2:

{

cout<<"1) Anti-Clockwise"<<endl;

cout<<"2) Clockwise"<<endl;

cin>>ch;

int angle;

switch(ch)

{

case 1:

{

cout<<"Enter angle of rotation in degree"<<endl;

cin>>angle;

t[0].x=cos(angle \*3.14/180); t[0].y=sin(angle \*3.14/180); t[0].z=0;

t[1].x=-sin(angle \*3.14/180); t[1].y=cos(angle \*3.14/180); t[1].z=0;

t[2].x=0; t[2].y=0; t[2].z=1;

cout<<endl;

for(int i=0;i<3;i++)

cout<<t[i].x<<" "<<t[i].y<<" "<<t[i].z<<endl;

break;

}

case 2:

{

cout<<"Enter angle of rotation"<<endl;

cin>>angle;

t[0].x=cos(angle \*3.14/180); t[0].y=-sin(angle \*3.14/180); t[0].z=0;

t[1].x=sin(angle \*3.14/180); t[1].y=cos(angle \*3.14/180); t[1].z=0;

t[2].x=0; t[2].y=0; t[2].z=1;

break;

}

}

break;

}

case 3:

{

cout<<"Note:Enter value 1 if you don't want scaling in a perticular direction'"<<endl;

cout<<"Enter x and y scaling factors respectively"<<endl;

cin>>t[0].x; t[0].y=0; t[0].z=0;

t[1].x=0; cin>>t[1].y; t[1].z=0;

t[2].x=0; t[2].y=0; t[2].z=1;

break;

}

case 4:

{

cout<<"Note:Enter value 0 if you don't want translation in a perticular direction'"<<endl;

cout<<"Enter x and y shearing factors respectively"<<endl;

t[0].x=1; t[0].y=0; t[0].z=0;

t[1].x=0; t[1].y=1; t[1].z=0;

cin>>t[2].x; cin>>t[2].y; t[2].z=1;

break;

}

case 5:

{

cout<<"Note:Enter value 0 if you don't want shearing in a perticular direction'"<<endl;

cout<<"Enter y and x shearing factors respectively"<<endl;

t[0].x=1; cin>>t[0].y; t[0].z=0;

cin>>t[1].x; t[1].y=1; t[1].z=0;

t[2].x=0; t[2].y=0; t[2].z=1;

break;

}

default:

{

cout<<"Wrong Choice Entered"<<endl;

break;

}

}

}

void transform()

{

for(int i=0;i<points;i++)

{

p1[0].x = p[i].x\*t[0].x + p[i].y\*t[1].x + p[i].z\*t[2].x;

p1[0].y = p[i].x\*t[0].y + p[i].y\*t[1].y + p[i].z\*t[2].y;

p1[0].z = p[i].x\*t[0].z + p[i].y\*t[1].z + p[i].z\*t[2].z;

p[i].x = (int)(p1[0].x);

p[i].y = (int)(p1[0].y);

p[i].z = (int)(p1[0].z);

}

setcolor(BLUE);

mat();

plot();

}

};

int main()

{

initwindow(800,800);

matrix m;

m.axis();

m.read();

m.trans\_mat();

m.transform();

getch();

return 0;

}

**Output:**



