NAME- Avinash Singh Bohra

COURSE- BCA

ROLL NO- 2092010

SUBJECT- Computer graphics lab

PRACTICLE-10

**OBJECTIVE-** TO PERFORM A 2D TRANSFORMATIONS.

**SYNTAX :-**

#include <iostream>

#include <math.h>

#include <graphics.h>

using namespace std;

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);

}

void translateTriangle(int &x1, int &y1, int &x2, int &y2, int &x3, int &y3, int tx, int ty)

{

x1 += tx;

y1 += ty;

x2 += tx;

y2 += ty;

x3 += tx;

y3 += ty;

}

void rotateTriangle(int &x1, int &y1, int &x2, int &y2, int &x3, int &y3, float angle)

{

float radians = angle \* 3.14159 / 180;

float cosVal = cos(radians);

float sinVal = sin(radians);

int tempX1 = x1;

int tempX2 = x2;

int tempX3 = x3;

int tempY1 = y1;

int tempY2 = y2;

int tempY3 = y3;

x1 = tempX1 \* cosVal - tempY1 \* sinVal;

y1 = tempX1 \* sinVal + tempY1 \* cosVal;

x2 = tempX2 \* cosVal - tempY2 \* sinVal;

y2 = tempX2 \* sinVal + tempY2 \* cosVal;

x3 = tempX3 \* cosVal - tempY3 \* sinVal;

y3 = tempX3 \* sinVal + tempY3 \* cosVal;

}

void scaleTriangle(int &x1, int &y1, int &x2, int &y2, int &x3, int &y3, float sx, float sy)

{

x1 \*= sx;

y1 \*= sy;

x2 \*= sx;

y2 \*= sy;

x3 \*= sx;

y3 \*= sy;

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int x1 = 2, y1 = 3, x2 = 70, y2 = 150, x3 = 120, y3 = 60;

int originalx1 = 2, originaly1 = 3, originalx2 = 70, originaly2 = 150, originalx3 = 120, originaly3 = 60;

int choice;

int tx, ty;

float angle;

float sx, sy;

int originx= getmaxx()/2;

int originy= getmaxy()/2;

outtextxy(originx, originy, "(0, 0)");

while (true)

{

cleardevice();

x1= originalx1;

x2= originalx2;

x3= originalx3;

y1= originaly1;

y2= originaly2;

y3= originaly3;

setcolor(WHITE);

line(getmaxx() / 2, 0, getmaxx() / 2, getmaxy());

line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);

drawTriangle(originx+ originalx1, originy- originaly1, originx+ originalx2, originy- originaly2, originx+ originalx3, originy- originaly3);

cout << "\nMenu:";

cout << "\n1. Translate Triangle";

cout << "\n2. Rotate Triangle";

cout << "\n3. Scale Triangle";

cout << "\n4. Exit";

cout << "\nEnter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

cout << "\nEnter translation factors (tx, ty): ";

cin >> tx >> ty;

translateTriangle(x1, y1, x2, y2, x3, y3, tx, ty);

break;

case 2:

cout << "\nEnter rotation angle: ";

cin >> angle;

rotateTriangle(x1, y1, x2, y2, x3, y3, angle);

break;

case 3:

cout << "\nEnter scaling factors (sx, sy): ";

cin >> sx >> sy;

scaleTriangle(x1, y1, x2, y2, x3, y3, sx, sy);

break;

case 4:

break;

default:

cout << "\nInvalid choice!";

}

setcolor(YELLOW);

if (x1>=0)x1+=originx; else x1-=originx;

if (y1>=0)y1=originy-y1; else y1+=originy;

if (x2>=0)x2+=originx; else x2-=originx;

if (y2>=0)y2=originy-y2; else y2+=originy;

if (x3>=0)x3+=originx; else x3-=originx;

if (y3>=0)y3=originy-y3; else y3+=originy;

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

delay(10000);

}

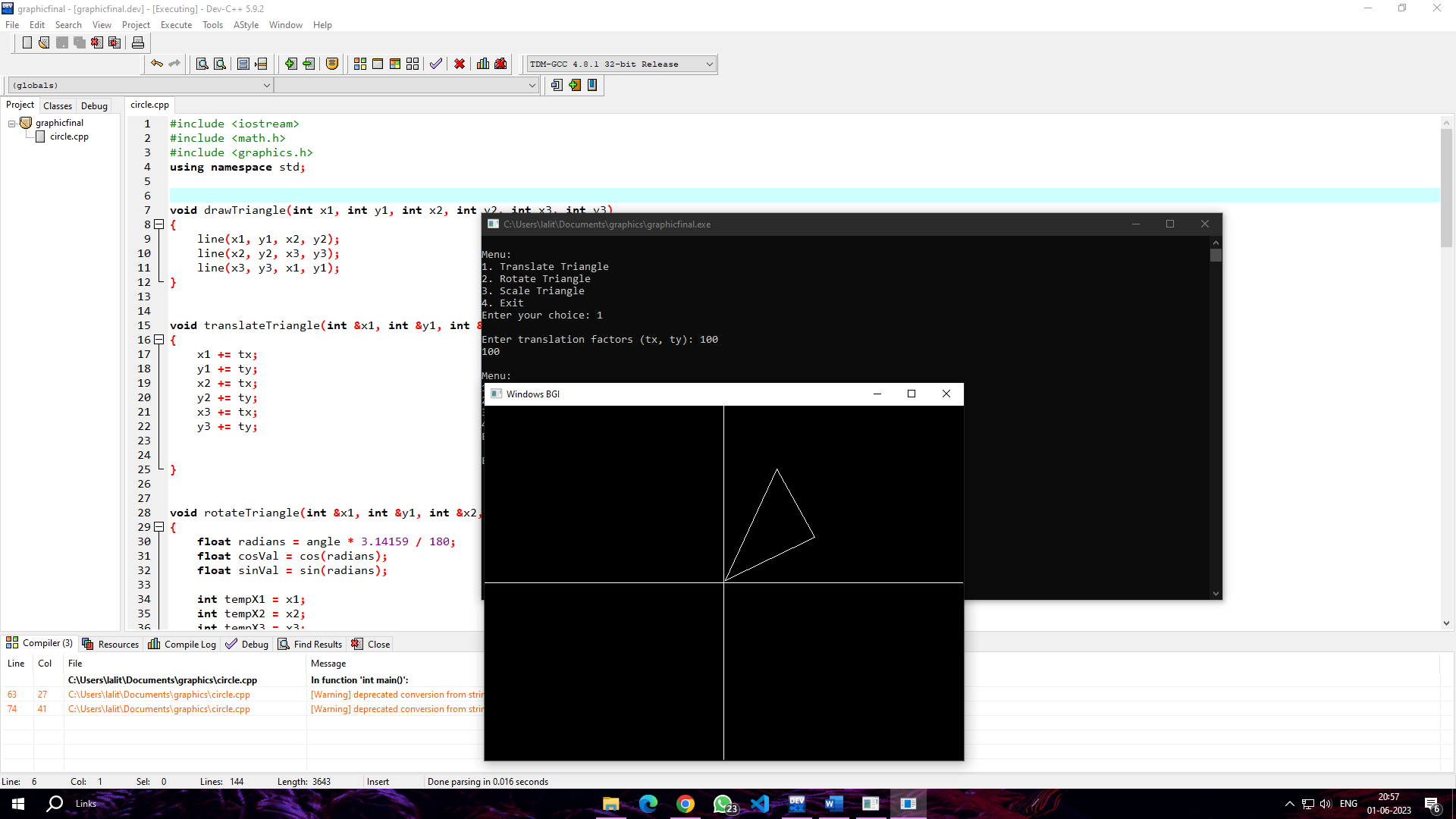
getch();

closegraph();

return 0;

}

**OUTPUT:**

****