**ARYAMAN MISHRA**

**19BCE1027**

#include <stdio.h>

#include<graphics.h>

#include<stdlib.h>

#include<conio.h>

#include<math.h>

void DrawFn();

void translate();

int graDriver=DETECT,graMode;

int n,xs[100],ys[100],i,xshift,yshift;

void DrawFn()

{

for(i=0;i<n;i++)

{

line(xs[i],ys[i],xs[(i+1)%n],ys[(i+1)%n]);

}

}

void translate()

{

for(i=0;i<n;i++)

{

xs[i]+=xshift;

ys[i]+=yshift;

}

}

void refx(int x1,int x2,int x3,int y1,int y2,int y3){

line(320,0,320,430);

line(0,240,640,240);

x1=(320-x1)+320;

x2=(320-x2)+320;

x3=(320-x3)+320;

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

}

void refy(int x1,int x2,int x3,int y1,int y2,int y3){

line(320,0,320,430);

line(0,240,640,240);

y1=(240-y1)+240;

y2=(240-y2)+240;

y3=(240-y3)+240;

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

}

void findNewCoordinate(int s[][2], int p[][1])

{

int temp[2][1] = { 0 };

for (int i = 0; i < 2; i++)

for (int j = 0; j < 1; j++)

for (int k = 0; k < 2; k++)

temp[i][j] += (s[i][k] \* p[k][j]);

p[0][0] = temp[0][0];

p[1][0] = temp[1][0];

}

void scale(int x[], int y[], int sx, int sy)

{

// Triangle before Scaling

line(x[0], y[0], x[1], y[1]);

line(x[1], y[1], x[2], y[2]);

line(x[2], y[2], x[0], y[0]);

// Initializing the Scaling Matrix.

int s[2][2] = { sx, 0, 0, sy };

int p[2][1];

// Scaling the triangle

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

{

p[0][0] = x[i];

p[1][0] = y[i];

findNewCoordinate(s, p);

x[i] = p[0][0];

y[i] = p[1][0];

}

// Triangle after Scaling

line(x[0], y[0], x[1], y[1]);

line(x[1], y[1], x[2], y[2]);

line(x[2], y[2], x[0], y[0]);

}

int main () {

char choice;

printf("Enter 1 for translation,2 for reflection,3 for rotation,4 for scaling,5 for shearing along x axis,6 for shearing along y axis.\n");

scanf("%c", &choice)

switch(choice) {

case '1' :

int graDriver=DETECT,graMode;

int n,xs[100],ys[100],i,xshift,yshift;

printf("Enter number of sides of polygon: ");

scanf("%d",&n);

printf("Enter co-rdinates: x,y for each vertex ");

for(i=0;i<n;i++)

scanf("%d%d",&xs[i],&ys[i]);

printf("Enter distances for translation (in x and y directions): ");

scanf("%d%d",&xshift,&yshift);

initgraph(&graDriver,&graMode,"C:\\TURBOC3\\BGI\\");

cleardevice();

//drawing original polygon in RED color

setcolor(RED);

DrawFn();

//Doing translation

translate();

//drawing translated polygon in BLUE color

setcolor(BLUE);

DrawFn();

getch();

break;

case '2' :

int gd=DETECT,gm;

int x1,y1,x2,y2,x3,y3;

clrscr();

initgraph(&gd,&gm,"c://turboc3//bgi");

line(320,0,320,430);

line(0,240,640,240);

x1=150;y1=100;

x2=220;y2=220;

x3=220;y3=110;

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

getch();

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

getch();

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

getch();

closegraph();

break;

case '3' :

intgd=0,gm,x1,y1,x2,y2,x3,y3;

double s,c, angle;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

setcolor(RED);

printf("Enter coordinates of triangle: ");

scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2, &x3, &y3);

setbkcolor(WHITE);

cleardevice();

line(x1,y1,x2,y2);

line(x2,y2, x3,y3);

line(x3, y3, x1, y1);

getch();

setbkcolor(BLACK);

printf("Enter rotation angle: ");

scanf("%lf", &angle);

setbkcolor(WHITE);

c = cos(angle \*M\_PI/180);

s = sin(angle \*M\_PI/180);

x1 = floor(x1 \* c + y1 \* s);

y1 = floor(-x1 \* s + y1 \* c);

x2 = floor(x2 \* c + y2 \* s);

y2 = floor(-x2 \* s + y2 \* c);

x3 = floor(x3 \* c + y3 \* s);

y3 = floor(-x3 \* s + y3 \* c);

cleardevice();

line(x1, y1 ,x2, y2);

line(x2,y2, x3,y3);

line(x3, y3, x1, y1);

getch();

closegraph();

case '4' :

int x[] = { 100, 200, 300 };

int y[] = { 200, 100, 200 };

int sx = 2, sy = 2;

int gd, gm;

detectgraph(&gd, &gm);

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

scale(x, y, sx,sy);

getch();

break;

case '5' :int gd=DETECT,gm;

int x,y,x1,y1,x2,y2,x3,y3,shear\_f;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("\n please enter first coordinate = ");

scanf("%d %d",&x,&y);

printf("\n please enter second coordinate = ");

scanf("%d %d",&x1,&y1);

printf("\n please enter third coordinate = ");

scanf("%d %d",&x2,&y2);

printf("\n please enter last coordinate = ");

scanf("%d %d",&x3,&y3);

printf("\n please enter shearing factor x = ");

scanf("%d",&shear\_f);

cleardevice();

line(x,y,x1,y1);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x,y);

setcolor(RED);

x=x+ y\*shear\_f;

x1=x1+ y1\*shear\_f;

x2=x2+ y2\*shear\_f;

x3=x3+ y3\*shear\_f;

line(x,y,x1,y1);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x,y);

getch();

closegraph();

break;

case '6':int gd=DETECT,gm;

int x,y,x1,y1,x2,y2,x3,y3,shear\_f;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("\n please enter first coordinate = ");

scanf("%d %d",&x,&y);

printf("\n please enter second coordinate = ");

scanf("%d %d",&x1,&y1);

printf("\n please enter third coordinate = ");

scanf("%d %d",&x2,&y2);

printf("\n please enter last coordinate = ");

scanf("%d %d",&x3,&y3);

printf("\n please enter shearing factor y = ");

scanf("%d",&shear\_f);

cleardevice();

line(x,y,x1,y1);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x,y);

setcolor(RED);

y=y+ x\*shear\_f;

y1=y1+ x1\*shear\_f;

y2=y2+ x2\*shear\_f;

y3=y3+ x3\*shear\_f;

line(x,y,x1,y1);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x,y);

getch();

closegraph();

break;

default :

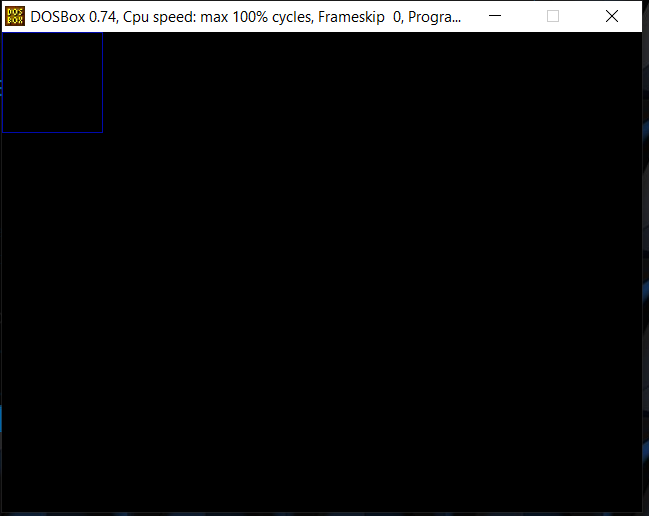
printf("Wrong Choice.Try Again.\n" );

}

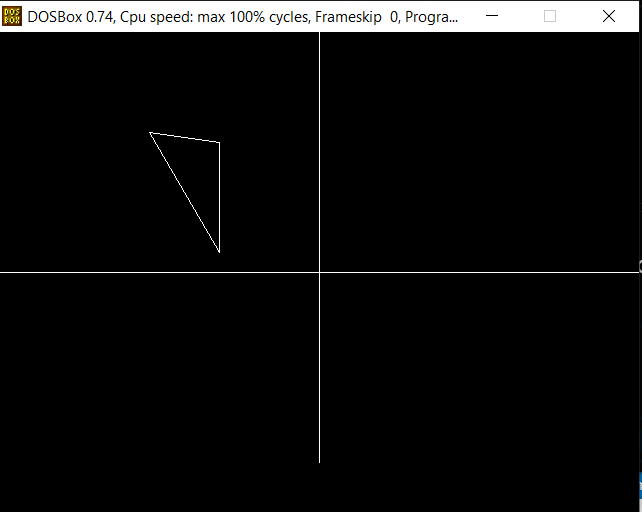
return 0;

}

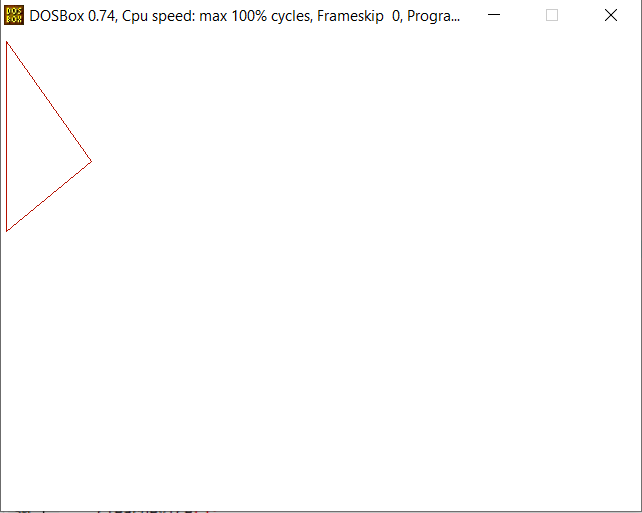
TRANSLATION:



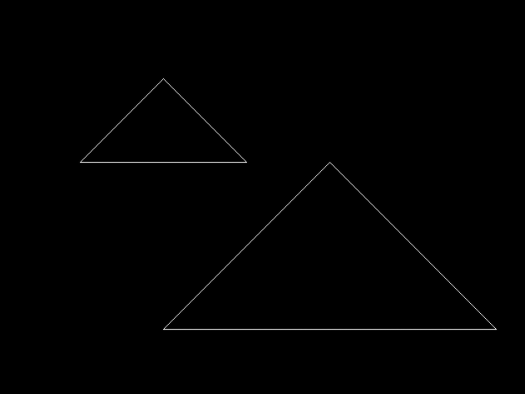
REFLECTION:



ROTATION:



SCALING:



SHEARING ALONG X-AXIS:



SHEARING ALONG Y-AXIS:

