NAME : VAIBHAV DNYANESHWAR MOHITE

ROLLNO: 25045

CLASS: SE IT A

# ASSIGNMENT NO - 6

CODE:

#include<iostream>

#include<stdio.h>

#include<GL/glut.h>

#include<stdlib.h> #include<math.h>

using namespace std;

static int menu\_id,submenu\_id; float x1,yy1,x2,y2,x3,y3,tx,ty,ABC[3][2];

void init(void); void display(void); void draw\_original(); void menu(int); void translation(); void scaling(); void rotation(); void shearing();

void reflection();

int main(int argc,char \*argv[])

{ x1=50; yy1=150;

x2=400; y2=100;

x3=200;

y3=346.41;

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(1000,1000);

glutInitWindowPosition(100,100); glutCreateWindow("2D-TRANSFORMATIONS"); init();

glutDisplayFunc(display); submenu\_id=glutCreateMenu(menu); glutAddMenuEntry("------MENU------",0); glutAddMenuEntry("1)TRANSLATION",1); glutAddMenuEntry("2)SCALING",2); glutAddMenuEntry("3)ROTATION",3); glutAddMenuEntry("4)SHEARING",4); glutAddMenuEntry("5)REFLECTION",5); menu\_id=glutCreateMenu(menu); glutAddSubMenu("DRAW",submenu\_id); glutAddMenuEntry("EXIT",6);

glutAttachMenu(GLUT\_RIGHT\_BUTTON); glutMainLoop();

return(0);

}

void init(void) {

glClearColor(0.0,0.0,0.0,1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-1000.0,1000.0,-1000.0,1000.0);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(1.0,1.0,1.0); glBegin(GL\_LINES); glVertex2i(-1000,0); glVertex2i(1000,0); glVertex2i(0,-1000); glVertex2i(0,1000); glEnd(); glFlush();

draw\_original();

}

void draw\_original()

{

glColor3f(1.0,0.0,0.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(x1,yy1); glVertex2f(x2,y2); glVertex2f(x3,y3); glEnd();

glFlush();

}

void menu(int key)

{ switch(key) { case 1: translation();

break;

case 2: scaling();

break;

case 3:

rotation(); break;

case 4: shearing();

break;

case 5: reflection(); break;

case 6: exit(0);

}

}

void translation()

{

printf("\n"); printf("Enter the translation factor tx:"); scanf("%f",&tx);

printf("\n"); printf("Enter the translation factor ty:"); scanf("%f",&ty);

ABC[0][0]=x1;

ABC[0][1]=yy1;

ABC[1][0]=x2;

ABC[1][1]=y2;

ABC[2][0]=x3;

ABC[2][1]=y3;

ABC[0][0]=ABC[0][0]+tx;

ABC[0][1]=ABC[0][1]+ty;

ABC[1][0]=ABC[1][0]+tx;

ABC[1][1]=ABC[1][1]+ty;

ABC[2][0]=ABC[2][0]+tx;

ABC[2][1]=ABC[2][1]+ty;

glColor3f(1.0,1.0,0.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

}

void scaling()

{

float sx,sy;

printf("\n"); printf("Enter the scaling factor sx:");

scanf("%f",&sx);

printf("\n"); printf("Enter the scaling factor sy:");

scanf("%f",&sy);

ABC[0][0]=x1;

ABC[0][1]=yy1;

ABC[1][0]=x2;

ABC[1][1]=y2;

ABC[2][0]=x3;

ABC[2][1]=y3;

ABC[0][0]=x1\*sx;

ABC[0][1]=yy1\*sy;

ABC[1][0]=x2\*sx;

ABC[1][1]=y2\*sy;

ABC[2][0]=x3\*sx;

ABC[2][1]=y3\*sy;

glColor3f(0.0,1.0,0.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

}

void rotation()

{

float angle,x,y;

int choice;

printf("\n"); printf("Enter the angle for rotation:");

scanf("%f",&angle);

x=0.01745\*angle;

begin:printf("\n");

printf("1)ANTI-CLOCKWISE\n"); printf("2)CLOCKWISE\n"); printf("Enter your choice:");

scanf("%d",&choice);

ABC[0][0]=x1;

ABC[0][1]=yy1;

ABC[1][0]=x2;

ABC[1][1]=y2;

ABC[2][0]=x3;

ABC[2][1]=y3;

switch(choice) { case 1:

ABC[0][0]=(x1\*cos(x)-yy1\*sin(x));

ABC[0][1]=(x1\*sin(x)+yy1\*cos(x));

ABC[1][0]=(x2\*cos(x)-y2\*sin(x));

ABC[1][1]=(x2\*sin(x)+y2\*cos(x));

ABC[2][0]=(x3\*cos(x)-y3\*sin(x));

ABC[2][1]=(x3\*sin(x)+y3\*cos(x));

glColor3f(0.0,0.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

break;

case 2:

ABC[0][0]=(x1\*cos(x)+yy1\*sin(x));

ABC[0][1]=(yy1\*cos(x)-x1\*sin(x));

ABC[1][0]=(x2\*cos(x)+y2\*sin(x));

ABC[1][1]=(y2\*cos(x)-x2\*sin(x));

ABC[2][0]=(x3\*cos(x)+y3\*sin(x));

ABC[2][1]=(y3\*cos(x)-x3\*sin(x));

glColor3f(0.0,0.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

break;

default: printf("\n"); printf("Enter valid choice!!!\n"); goto begin;

}

}

void shearing() {

float shx,shy;

printf("\n"); printf("Enter shearing factor shx:");

scanf("%f",&shx);

printf("\n"); printf("Enter shearing factor shy:");

scanf("%f",&shy);

ABC[0][0]=x1;

ABC[0][1]=yy1;

ABC[1][0]=x2;

ABC[1][1]=y2;

ABC[2][0]=x3;

ABC[2][1]=y3;

ABC[0][0]=abs(ABC[0][0]+shx\*yy1);

ABC[0][1]=abs(ABC[0][1]+shy\*x1);

ABC[1][0]=abs(ABC[1][0]+shx\*y2); ABC[1][1]=abs(ABC[1][1]+shy\*x2);

ABC[2][0]=abs(ABC[2][0]+shx\*y3);

ABC[2][1]=abs(ABC[2][1]+shy\*x3);

glColor3f(1.0,0.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

}

void reflection()

{

int choice;

start:printf("\n"); printf("1)Reflection about X-axis\n"); printf("2)Reflection about Y-axis\n"); printf("3)Reflection about the line y=x\n"); printf("4)Reflection about Origin\n"); printf("Enter your choice:");

scanf("%d",&choice);

ABC[0][0]=x1;

ABC[0][1]=yy1;

ABC[1][0]=x2;

ABC[1][1]=y2;

ABC[2][0]=x3;

ABC[2][1]=y3;

switch(choice) { case 1:

ABC[0][0]=x1;

ABC[0][1]=-yy1;

ABC[1][0]=x2;

ABC[1][1]=-y2;

ABC[2][0]=x3;

ABC[2][1]=-y3;

glColor3f(0.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd();

glFlush();

break;

case 2:

ABC[0][0]=-x1;

ABC[0][1]=yy1;

ABC[1][0]=-x2;

ABC[1][1]=y2;

ABC[2][0]=-x3;

ABC[2][1]=y3;

glColor3f(0.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

break;

case 3:

ABC[0][0]=yy1;

ABC[0][1]=x1;

ABC[1][0]=y2;

ABC[1][1]=x2;

ABC[2][0]=y3;

ABC[2][1]=x3;

glColor3f(0.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

break;

case 4:

ABC[0][0]-x1;

ABC[0][1]=-yy1;

ABC[1][0]=-x2;

ABC[1][1]=-y2;

ABC[2][0]=-x3;

ABC[2][1]=-y3;

glColor3f(0.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(ABC[0][0],ABC[0][1]); glVertex2f(ABC[1][0],ABC[1][1]); glVertex2f(ABC[2][0],ABC[2][1]);

glEnd(); glFlush();

break;

default: printf("\n"); printf("Enter valid choice!!!\n");

goto start;

}

}

OUTPUT:

