#include<stdio.h>

#include<GL/glut.h>

#include<stdlib.h>

#include<math.h>

int ax,ay,wid,ang,sx,sy;

float a[3][3],b[3][3],c[3][3];

int sign(int x)

{

if(x>0) return 1;

if(x<0) return -1;

if(x==0) return 0;

}

void dis(float a[3][3])

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0,1.0,0.0);

int i;

glColor3f(1.0,1.0,0.0);

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

{

glBegin(GL\_LINES);

glVertex2i(a[i][0],a[i][1]);

glVertex2i(a[(i+1)%3][0],a[(i+1)%3][1]);

glEnd();

glFlush();

}

}

void dis1(float c[3][3])

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0,1.0,0.0);

int i;

glColor3f(1.0,1.0,0.0);

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

{

glBegin(GL\_LINES);

glVertex2i(c[i][0],c[i][1]);

glVertex2i(c[(i+1)%3][0],c[(i+1)%3][1]);

glEnd();

glFlush();

}

}

void rot(float b[3][3])

{

int i,j,s1;

float t;

if(ang<0)

s1=-1;

else

s1=1;

t=ang\*3.14/180;

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

{

for(j=0;j<3;j++)

{

if(i==j)

b[i][j]=1;

else

b[i][j]=0;

}

}

b[0][0]=b[1][1]=cos(t);

b[0][1]=s1\*sin(t);

b[1][0]=-s1\*sin(t);

}

void sca(float b[3][3])

{

int i,j,s1;

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

{

for(j=0;j<3;j++)

{

if(i==j)

b[i][j]=1;

else

b[i][j]=0;

}

}

b[0][0]=sx;

b[1][1]=sy;

}

void tra(float b[3][3])

{

int i,j,s1;

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

{

for(j=0;j<3;j++)

{

if(i==j)

b[i][j]=1;

else

b[i][j]=0;

}

}

b[2][0]=sx;

b[2][1]=sy;

}

void shr(float b[3][3])

{

int i,j,s1;

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

{

for(j=0;j<3;j++)

{

if(i==j)

b[i][j]=1;

else

b[i][j]=0;

}

}

b[0][1]=sy;

b[1][0]=sx;

}

void mul(float a[3][3],float b[3][3],float c[10][3])

{

int i,j,k;

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

{

for(j=0;j<3;j++)

{

c[i][j]=0;

for(k=0;k<3;k++)

{

c[i][j]+=a[i][k]\*b[k][j];

}

}

}

}

void init(void)

{

glClearColor(0.0,0.0,0.0,0.0);

gluOrtho2D(-600.0,600.0,-600.0,600.0);

}

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

{

int ch;

int bx,by,cx,cy,mx,my,height,wi,abc;

printf("\n Enter Vertex 1:");

scanf("%d%d",&ax,&ay);

printf("\n Enter width:");

scanf("%d",&wid);

bx=ax+wid;

by=ay;

height=(sqrt(3)\*wid )/2;

mx=(ax+bx)/2;

my=ay;

cx=mx;

cy=my+height;

a[0][0]=ax; a[0][1]=ay; a[0][2]=1;

a[1][0]=bx; a[1][1]=by;a[1][2]=1;

a[2][0]=cx; a[2][1]=cy;a[2][2]=1;

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(800,800); //window size is imp...

glutInitWindowPosition(100,100);

glutCreateWindow("translation");

init();

glClear(GL\_COLOR\_BUFFER\_BIT);

for(;;)

{

printf("\n1.original\n2.Rotate\n3.Scaling\n4.Translate\n5.Shear\n6.Exit\nEnter your choice");

scanf("%d",&ch);

switch(ch)

{

case 1:

dis(a);

break;

case 2:

printf("\n Enter angle of rotation:");

scanf("%d",&ang);

rot(b);

mul(a,b,c);

//dis(a);

dis(c);

break;

case 3:

printf("\n Enter scaling ratio");

scanf("%d%d",&sx,&sy);

sca(b);

mul(a,b,c);

//dis(a);

dis(c);

break;

case 4:

printf("\n Enter translation ratio");

scanf("%d%d",&sx,&sy);

tra(b);

mul(a,b,c);

//dis(a);

dis(c);

break;

case 5:

printf("\n Enter shear ratio");

scanf("%d%d",&sx,&sy);

shr(b);

mul(a,b,c);

//dis(a);

dis(c);

break;

case 6:

exit(0);

}

}

glFlush();

glutMainLoop();

return 0;

}