**Write a program for implementation of Reflection Transformation.**

#include<iostream.h>

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<dos.h>

void main()

{

int gd=DETECT,gm;

void drawaxis(void);

void dda(int,int,int,int);

void polygon(int,int,int,int);

void ref();

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

ref();

getch();

closegraph();

}

void drawaxis()

{

int i;

char s[25];

line(320,0,320,480);

outtextxy(5,230,"X-AXIS");

settextstyle(0,1,1);

outtextxy(336,6,"Y-AXIS");

line(0,240,640,240);

settextstyle(0,0,0);

for(i=240;i>-240;i-=20)

{

outtextxy(318,237-i,"-");

if(i==0)

continue;

itoa(i,s,10);

outtextxy(286,240-i,s);

}

for(i=-300;i<300;i+=20)

{

itoa(i,s,10);

settextstyle(0,1,0);

outtextxy(i+325,245,s);

}

}

void dda(int x1,int y1,int x2,int y2)

{

int k,dx,dy,steps;

float x,y,incrx,incry;

dx=x2-x1;

dy=y2-y1;

if(abs(dx)>abs(dy))

steps=abs(dx);

else

steps=abs(dy);

incrx=(float)dx/steps;

incry=(float)dy/steps;

x=x1;

y=y1;

putpixel(320+x,240-y,WHITE);

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

{

x+=incrx;

y+=incry;

putpixel(320+x,240-y,WHITE);

}

}

void polygon(int cx[],int cy[],int n)

{

for(int i=0;i<n-1;i++)

{

dda(cx[i],cy[i],cx[i+1],cy[i+1]);

dda(cx[n-1],cy[n-1],cx[0],cy[0]);

}

}

void ref(void)

{

float h[2][2];

int ax[15],ay[15],n,i,j;

int ch,tx[15]={0},ty[15]={0},temp;

cout<<"\n\n\tEnter the sides of polygon";

cin>>n;

if(n<3)

{

cout<<"Invalid side"<<"\n";

getch();

exit(0);

}

cout<<"\tEnter co-ordinate of polygon \n";

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

{

cout<<"\t Enter for side"<<i+1<<" ";

cin>>ax[i]>>ay[i];

cleardevice();

}

while(1)

{

cleardevice();

cout<<"1 :Along x-axis \n";

cout<<"2 :Along y-axis \n";

cout<<"3 :Along x=y \n";

cout<<"4 :Along x=-y \n";

cout<<"5 :Along origin \n";

cout<<"6 :exit \n";

cout<<"Enter your choice ";

cin>>ch;

cleardevice();

drawaxis();

polygon(ax,ay,n);

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

{

for(j=0;j<2;j++)

{

if(i==j)

h[i][j]=1;

else

h[i][j]=0;

}

}

switch(ch)

{

case 1:

h[0][0]=-h[0][0];

break;

case 2:

h[1][1]=-h[1][1];

break;

case 3:

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

{

for(j=0;j<2;j++)

{

if(i==j)

h[i][j]=0;

else

h[i][j]=1;

}

}

break;

case 4:

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

{

for(j=0;j<2;j++)

{

if(i==j)

h[i][j]=0;

else

h[i][j]=1;

}

}

h[0][1]=-h[0][1];

h[1][0]=-h[1][0];

break;

case 5:

h[0][0]=-h[0][0];

h[1][0]=-h[1][1];

break;

default:

closegraph();

exit(1);

}

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

{

temp=ax[i]\*h[0][0]+ay[i]\*h[1][0];

ty[i]=(int)(ax[i]\*h[0][1]+ay[i]\*h[1][1]);

tx[i]=(int)temp;

}

sleep(2);

cleardevice();

drawaxis();

polygon(tx,ty,n);

getch();

}

}



















