**Practical No 7.B**

**Implementation of 2D shear and reflection transformations.**

**Aim: Write a program to implement 2D shear and reflection transformations.**

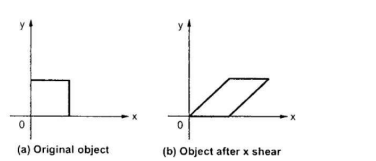
**Theory:**

**Shear**

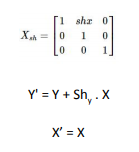
A transformation that slants the shape of an object is called the shear transformation. There are two shear transformations X-Shear and Y-Shear. One shifts X coordinates values and other shifts Y coordinate values. However; in both the cases only one coordinate changes its coordinates and other preserves its values. Shearing is also termed as Skewing.

**X-Shear**

The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.

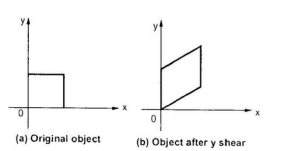


The transformation matrix for X-Shear can be represented as

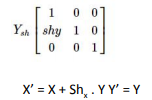


**Y-Shear**

The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.

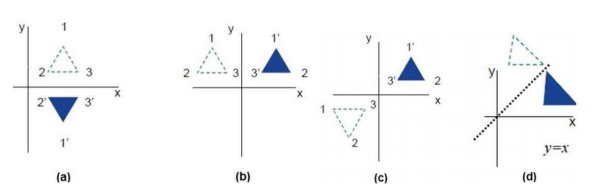


The Y-Shear can be represented in matrix from as:



**Reflection**

Reflection is the mirror image of original object. In other words, we can say that it is a rotation operation with 180°. In reflection transformation, the size of the object does not change. The following figures show reflections with respect to X and Y axes, and about the origin respectively.



**Conclusion: We have implemented 2D shear and reflection transformations.**

**Code:**

#include<iostream.h>

#include<process.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void disp(int n,float c[][3]) {

float maxx,maxy;

int i;

maxx=getmaxx();

maxy=getmaxy();

maxx=maxx/2;

maxy=maxy/2;

i=0;

while(i<n-1) {

line(maxx+c[i][0],maxy-c[i][1],maxx+c[i+1][0],maxy-c[i+1][1]);

i++; }

i=n-1;

line(maxx+c[i][0],maxy-c[i][1],maxx+c[0][0],maxy-c[0][1]);

setcolor(GREEN);

line(0,maxy,maxx\*2,maxy);

line(maxx,0,maxx,maxy\*2);

setcolor(WHITE); }

void mul(int n,float b[][3],float c[][3],float a[][3]) {

int i,j,k;

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

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

a[i][j]=0;

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

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

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

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

void reflection(int n,float c[][3]) {

float b[10][3],a[10][3];

int i=0,ch,j;

cleardevice();

cout<<"\n\t\* \* MENU \* \*";

cout<<"\n\t1) ABOUT X-AXIS";

cout<<"\n\t2) ABOUT Y-AXIS";

cout<<"\n\t3) ABOUT ORIGIN";

cout<<"\n\t4) ABOUT X=Y";

cout<<"\n\t5) ABOUT -X=Y";

cout<<"\n\t6) EXIT";

cout<<"\n\tENTER YOUR CHOICE : ";

cin>>ch;

clrscr();

cleardevice();

disp(n,c);

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

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

b[i][j]=0;

if(i==j)

b[i][j]=1; }

switch(ch){

case 1:

b[1][1]=-1;

break;

case 2:

b[0][0]=-1;

break;

case 3:

b[0][0]=-1;

b[1][1]=-1;

break;

case 4:

b[0][0]=0;

b[1][1]=0;

b[0][1]=1;

b[1][0]=1;

break;

case 5:

b[0][0]=0;

b[1][1]=0;

b[0][1]=-1;

b[1][0]=-1;

break;

case 6:

break;

default:

cout<<"\n\tINVALID CHOICE ! ";

break;

}

mul(n,b,c,a);

setcolor(RED);

disp(n,a);

}

void shearing(int n,float c[][3]) {

float b[10][3],sh,a[10][3];

int i=0,ch,j;

cleardevice();

cout<<"\n\t\* \* \* MENU \* \* \*";

cout<<"\n\t1) X SHEARING";

cout<<"\n\t2) Y SHEARING";

cout<<"\n\t3) EXIT ";

cout<<"\n\tENTER YOUR CHOICE : ";

cin>>ch;

if(ch==3)

return;

cout<<"\n\tENTER THE VALUE for SHEARING:";

cin>>sh;

clrscr();

cleardevice();

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

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

b[i][j]=0;

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

b[i][i]=1;

switch(ch) {

case 1:

b[1][0]=sh;

break;

case 2:

b[0][1]=sh;

break;

case 3:

break;

default:

cout<<"\n\tINVALID CHOICE ! ";

break; }

mul(n,b,c,a);

setcolor(RED);

disp(n,a); }

void main() {

int i,j,k,cho,n,gd=DETECT,gm;

float c[10][3],tx,ty,sx,sy,ra;

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

cout<<"\nEnter the number of vertices : ";

cin>>n;

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

cout<<"\nEnter the co-ordinates of the "<<i+1<<" vertex:";

cin>>c[i][0]>>c[i][1];

c[i][2]=1; }

do {

clrscr();

cleardevice();

cout<<"\n\t\t\t \* \* \* MENU \* \* \*";

cout<<"\n\t 1) REFLECTION ";

cout<<"\n\t 2) SHEARING";

cout<<"\n\t 3) EXIT";

cout<<"\n\t ENTER YOUR CHOICE:";

cin>>cho;

switch(cho) {

case 1:

clrscr();

cleardevice();

setcolor(BLUE);

disp(n,c);

reflection(n,c);

getch();

break;

case 2:

clrscr();

cleardevice();

setcolor(BLUE);

disp(n,c);

shearing(n,c);

getch();

break;

case 3 :

exit(0);

break;

default:

cout<<"\n\tInvalid choice !!";

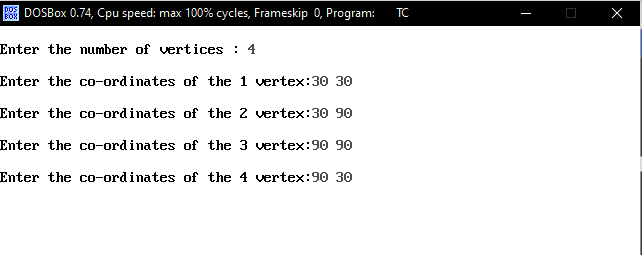
break; } }

while(cho!=3);

getch();

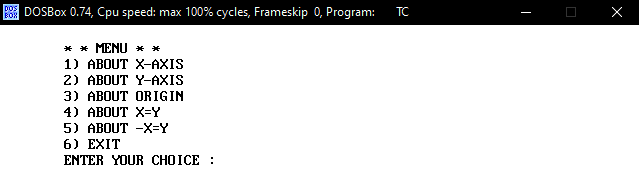
closegraph(); }

**Output:**

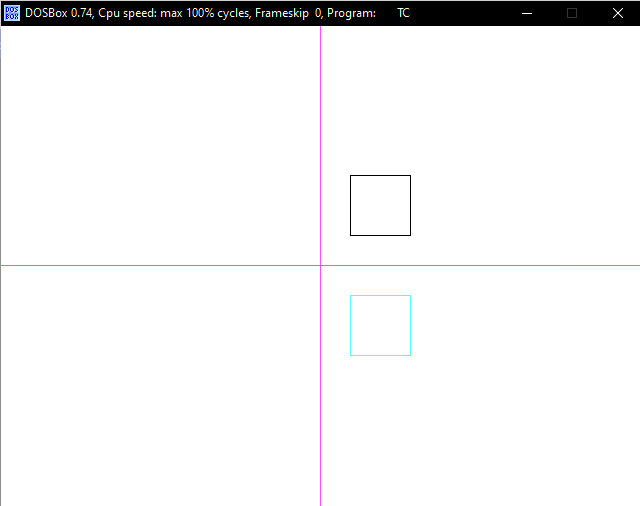


**REFLECTION:**

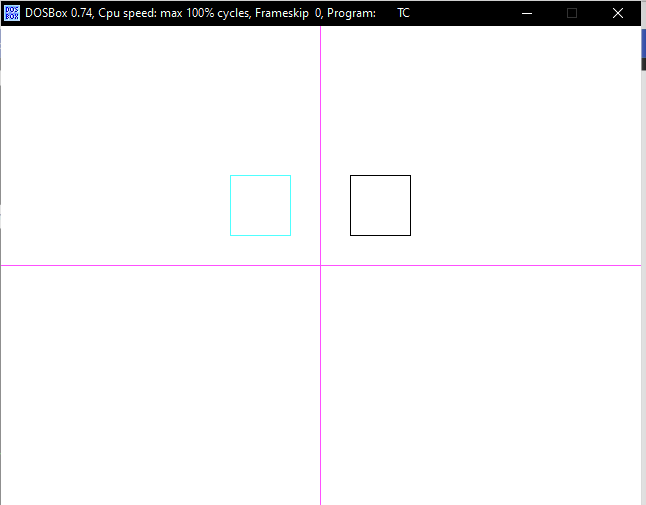




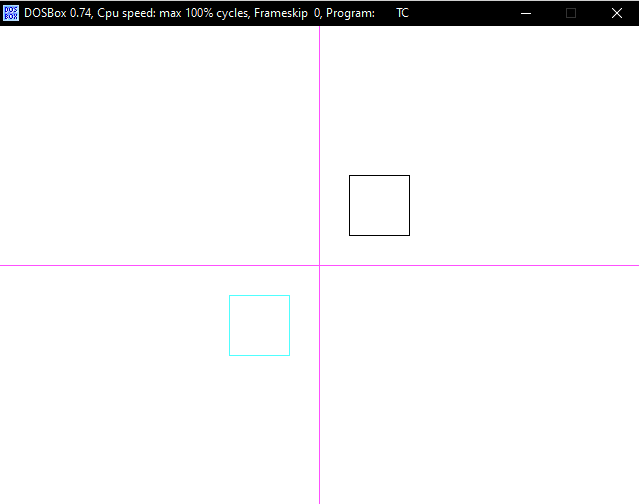
**1) X-AXIS:**



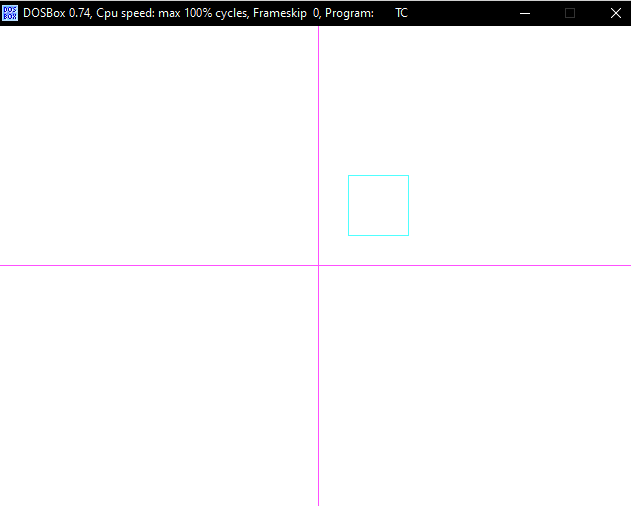
**2) Y-AXIS:**



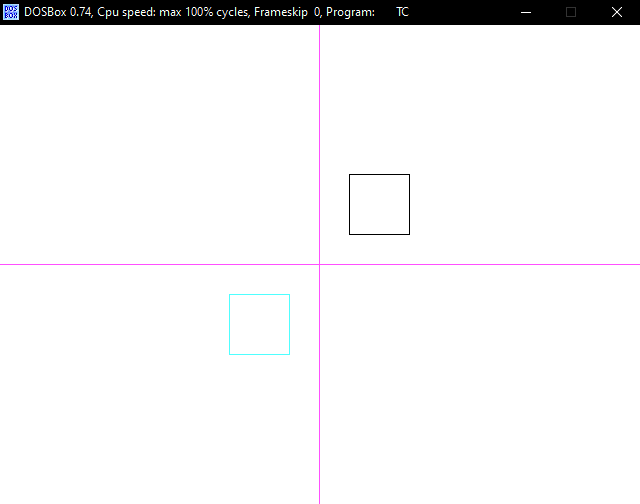
**3) ORIGIN:**



**4) X=Y:**



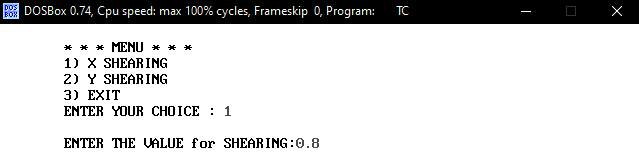
**5) -X=Y:**

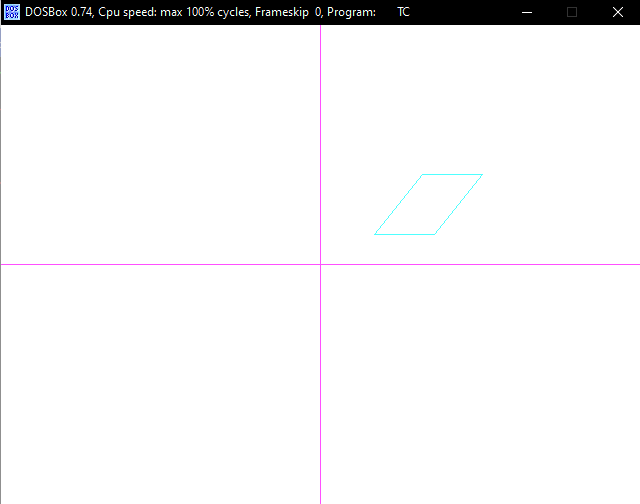


**SHEARING**



**1) X SHEARING:**





**2) Y SHEARING:**

