**Aim:**  To implement 2D Transformations: Translation, Scaling, Rotation.

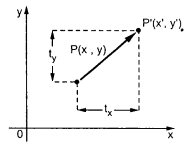
**Objective:**

To understand the concept of transformation, identify the process of transformation and application of these methods to different object and noting the difference between these transformations.

**Theory:**

**1) Translation –**

Translation is defined as moving the object from one position to another position along straight line path. We can move the objects based on translation distances along x and y axis. tx denotes translation distance along x-axis and ty denotes translation distance along y axis.

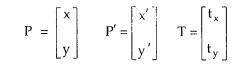


Consider (x,y) are old coordinates of a point. Then the new coordinates of that same point (x’,y’) can be obtained as follows:

**x’ = x + tx**

**y’ = y + ty**

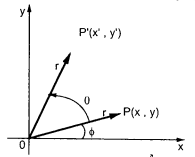
We denote translation transformation as P. we express above equations in matrix form as:  
P’ = P + T , where



**2) Rotation –**

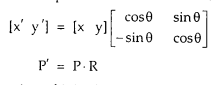
A rotation repositions all points in an object along a circular path in the plane centered at the

pivot point. We rotate an object by an angle theta. New coordinates after rotation depend on both x and y.





The above equations can be represented in the matrix form as given below



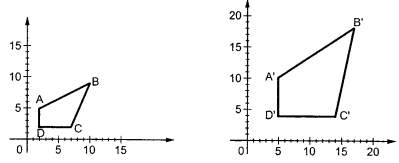
where R is the rotation matrix and it is given as



**3) Scaling -**

scaling refers to changing the size of the object either by increasing or decreasing. We will

increase or decrease the size of the object based on scaling factors along x and y-axis.



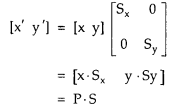
If (x, y) are old coordinates of object, then new coordinates of object after applying scaling

transformation are obtained as:

x’ = x \* Sx

y’ = y \* Sy

Sx and Sy are scaling factors along x-axis and y-axis. we express the above equations in matrix form as:



**Program:**

1]

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void main()

{

int gd=DETECT,gm,ch,tx,ty,nx1,nx2,ny1,ny2,sx,sy;

double r,t;

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

line(100,100,200,100);

printf("1.Transition\n2.ROTATION\n3.SCALING\n");

printf("ENTER YOUR CHOICE");

scanf("%d",&ch);

switch(ch)

{

case 1:printf("ENTER TRANSITION FACTOR: ");

scanf("%d%d",&tx,&ty);

nx1=100+tx;

ny1=100+ty;

nx2=200+tx;

ny2=100+ty;

line(nx1,ny1,nx2,ny2);

getch();

case 2:printf("ENTER ANGLE: ");

scanf("%lf",&r);

t=(3.14\*r)/180;

nx1=(int)(100+(100)\*cos(t)-(0)\*sin(t));

ny1=(int)(100+(100)\*sin(t)+(0)\*cos(t));

line(100,100,nx1,ny1);

getch();

case 3:printf("ENTER SCALING FACTOR: ");

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

nx1=100\*sx;

ny1=100\*sy;

nx2=200\*sx;

ny2=100\*sy;

line(nx1,ny1,nx2,ny2);

getch();

default: printf("INVALID");

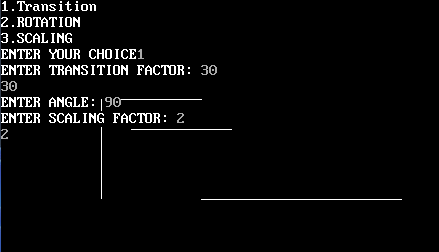
}

getch();

closegraph();

}

**Output -**



**Conclusion:** Comment on :

1. Application of transformation-3D Modeling and Animation,Image Editing,Data Visualization etc
2. Difference noted between methods-scaling modifies an object's size, translation changes its position, and rotation adjusts its orientation. These transformations are often used in combination to achieve more complex visual effects and animations in computer graphics.
3. Application t different object-translation-object movement

-rotational-orientation adjustment

-scaling-size modification