**Practical No. 10**

***Title****:-* Write a C program for 2D Rotation about an arbitrary Point.

|  |
| --- |
|  |

***Relevant Course Outcome(s):-***

Perform and demonstrate basic and composite graphical transformations on given object.

***Resources Required (Hardware & Softwares):-***

A Desktop PC/ Laptop

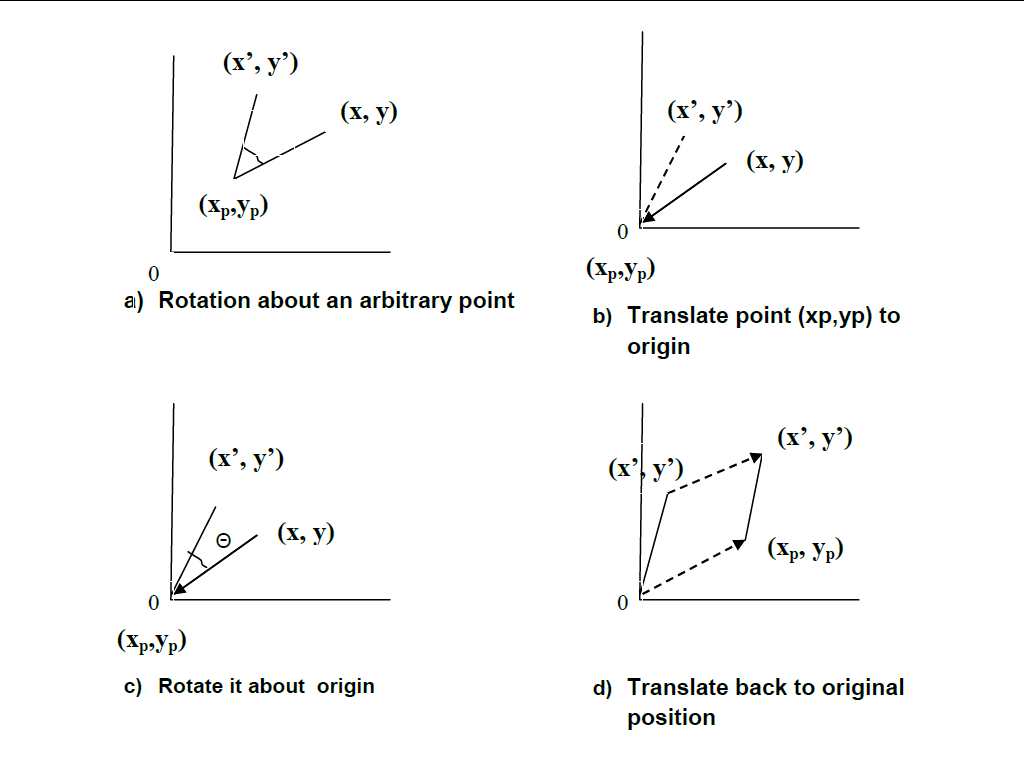
Ansi C/ Turbo C/ (Any distribution) installed

***Theory****:-*

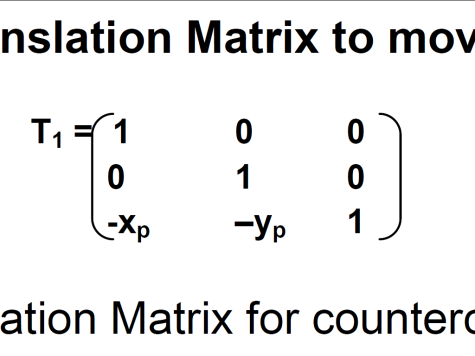
To do rotation of an object about any selected arbitrary point **(xp,yp)** we have to carry three steps

1. Translate point (xp,yp) to origin

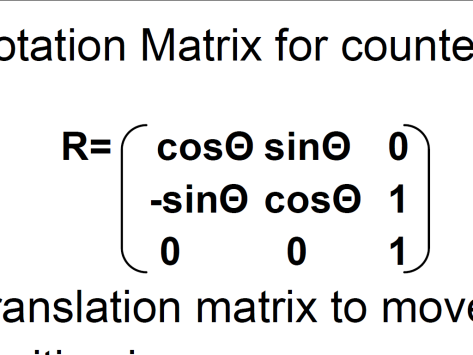
2. Rotate it about origin

3. Finally translate center of rotation where it belongs 

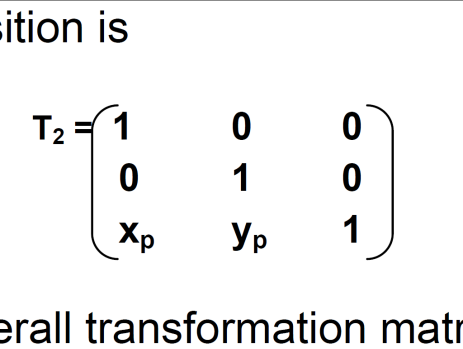
**Translation Matrix to move point (xp,yp) to origin is**

****

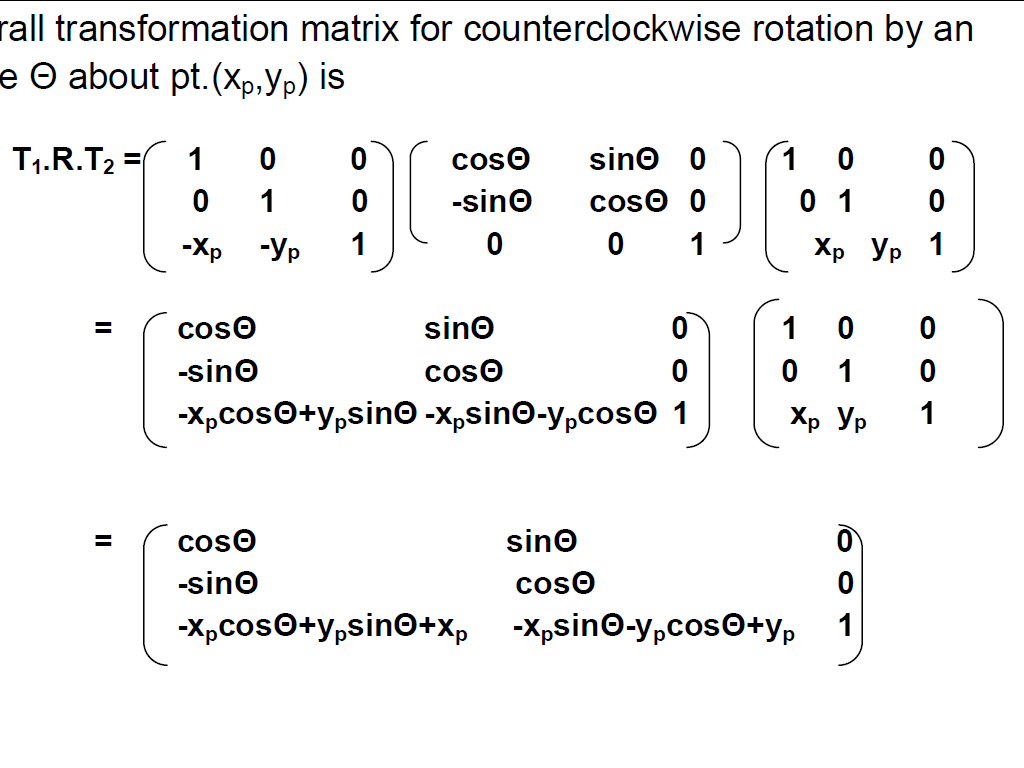
Rotation Matrix for counterclockwise rotation of pt. about origin



Translation matrix to move center point back to its original position is



Overall transformation matrix for counterclockwise rotation by an angle Θ about pt.(xp,yp) is



***Algorithm:-***

1. Read the coordinates of the point xp,yp about which rotation is to be performed.

2. Read the angle of rotation.

3. Read the coordinates of the polygon(say triangle)

4. Multiply the Translation, Rotation and Translation Inverse to obtain the resultant Rotation Matrix

5. Multiply the original coordinates of the triangle with the resultant Rotation Matrix to obtain transformed triangle

6. Draw the transformed polygon

7. Stop.

**Program for Rotation about an Arbitrary Point**

#include<graphics.h>

#include<stdio.h>

#include<math.h>

#include<dos.h>

#include<conio.h>

int gdriver=DETECT,gmode,j,k,sum,mult;

int pi[8];

double b[3][3]={1,0,0,

0,1,0,

0,0,1};

int c[1][1];

float a[1][1];

int x=0,y=0;

void matmul(float[8]);

void main()

{

int angle;

float p1[10]= {50,150,

80,150,

80,220,

50,220,

50,150,

};

float p2[10]= {50,150,

70,130,

100,130,

80,150,

50,150

};

float p3[10]= {80,150,

100,130,

100,200,

80,220,

80,150

};

detectgraph(&gdriver,&gmode);

initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi");

setcolor(4);

matmul(p1);

setfillstyle(1,4);

fillpoly(4,pi);

matmul(p2);

setfillstyle(1,1);

fillpoly(4,pi);

matmul(p3);

setfillstyle(1,15);

fillpoly(4,pi);

getch();

for(angle=0;angle<360;angle++)

{

setcolor(0);

matmul(p1);

setfillstyle(1,0);

fillpoly(4,pi);

matmul(p2);

setfillstyle(1,0);

fillpoly(4,pi);

matmul(p3);

setfillstyle(1,0);

fillpoly(4,pi);

b[0][0] =cos(angle\*3.142/180);

b[0][1] =sin(angle\*3.142/180);

b[1][0] =-sin(angle\*3.142/180);

b[1][1] =cos(angle\*3.142/180);

b[1][2] = 0;

b[2][0] =-x\*cos(angle\*3.142/180)+y\*sin(angle\*3.142/180)+x;

b[2][1] =-x\*sin(angle\*3.142/180)+y\*cos(angle\*3.142/180)+y;

b[2][2] = 0;

matmul(p1);

setfillstyle(1,4);

fillpoly(4,pi);

matmul(p2);

setfillstyle(1,1);

fillpoly(4,pi);

matmul(p3);

setfillstyle(1,15);

fillpoly(4,pi);

closegraph();

}

for(angle=360;angle>=0;angle--)

{

setcolor(0);

matmul(p1);

setfillstyle(1,0);

fillpoly(4,pi);

matmul(p2);

setfillstyle(1,0);

fillpoly(4,pi);

matmul(p3);

setfillstyle(1,0);

fillpoly(4,pi);

b[0][0] =cos(angle\*3.142/180);

b[0][1] =sin(angle\*3.142/180);

b[0][2] = 0;

b[1][0] =-sin(angle\*3.142/180);

b[1][1] =cos(angle\*3.142/180);

b[1][2] = 0;

b[2][0] =-x\*cos(angle\*3.142/180)+y\*sin(angle\*3.142/180)+x;

b[2][1] =-x\*sin(angle\*3.142/180)+y\*cos(angle\*3.142/180)+y;

b[2][2] = 0;

matmul(p1);

setfillstyle(1,4);

fillpoly(4,pi);

matmul(p2);

setfillstyle(1,1);

fillpoly(4,pi);

matmul(p3);

setfillstyle(1,15);

fillpoly(4,pi);

delay(100);

}

getch();

}

void matmul(float p[10])

{

int i;

for(i=0;i<9;i=i+2)

{ a[0][0]=p[i];

a[0][1]=p[i+1];

c[0][0]=a[0][0]\*b[0][0]+a[0][1]\*b[1][0]+b[2][0]+320;

c[0][1]=a[0][0]\*b[0][1]+a[0][1]\*b[1][1]+b[2][1]+240;

pi[i]=c[0][0];

pi[i+1]=c[0][1]; }

}

**Output:- ( Paste your own Output )**

**Conclusion:-** Thus,we have written and implement program for **Rotation about an arbitrary point**.