**Ashesh Kumar**

**501254**

**I.T. – 5**

**Q. Implement 2D Transformations – Translation, Rotation, Scaling .**

#include<graphics.h>

#include<conio.h>

#include<math.h>

int main()

{

int i, x, y, tx, ty, sx, sy, angle=10, xmax, ymax, xmid, ymid, op;

int gd,gm;

float p1[10]= { 50,50,

100,50,

100,100,

50,100,

50,50,

};

int pi[10];

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

0,1,0,

0,0,1

};

int c[1][1];

float a[1][1];

printf("\nSelect the transformation : ");

printf("\n1 : Traslation");

printf("\n2 : Rotation");

printf("\n3 : Scaling");

printf("\n4 : Rotation about arbitrary point");

printf( "\nEnter the option : ");

scanf("%d",&op);

switch(op)

{

case 1: printf("\nEnter x traslation : ");

scanf("%d",&tx);

printf("\nEnter y traslation : ");

scanf("%d",&ty);

b[0][0] = 1;

b[0][1] = 0;

b[0][2] = 0;

b[1][0] = 0;

b[1][1] = 1;

b[1][2] = 0;

b[2][0] = tx;

b[2][1] = ty;

b[2][2] = 1;

break;

case 2: printf("\nEnter Rotation angle : ");

scanf("%d",&angle);

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] = 0;

b[2][1] = 0;

b[2][2] = 1;

break;

case 3: printf("\nEnter x scaling : ");

scanf("%d",&sx);

printf("\nEnter y scaling : ");

scanf("%d",&sy);

b[0][0] = sx;

b[0][1] = 0;

b[0][2] = 0;

b[1][0] = 0;

b[1][1] = sy;

b[1][2] = 0;

b[2][0] = 0;

b[2][1] = 0;

b[2][2] = 1;

break;

case 4: printf("\nEnter x coordinate of arbitrary point : ");

scanf("%d",&x);

printf("\nEnter y coordinate of arbitrary point : ");

scanf("%d",&y);

printf("\nEnter Rotation angle : ");

scanf("%d",&angle);

tx = x;

ty = y;

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] = -tx\* cos(angle\*3.142/180) + ty\*sin(angle\*3.142/180)+tx;

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

b[2][2] = 1;

}

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"C:\\Turboc3\\BGI"); // Initialize graphics

xmax = getmaxx(); // Get maximum x coordinate

ymax = getmaxy(); // Get maximum y coordinate

xmid = xmax/2; // Get the center x coordinate

ymid = ymax/2; // Get the center y coordinate

setcolor(1);

line(xmid,0,xmid,ymax); // Draw y coordinate

line(0, ymid, xmax, ymid); // Draw x coordinate

setcolor(4);

for (i=0; i<8;i=i+2)

{

line(p1[i]+xmid,ymid-p1[i+1],xmid+p1[i+2],ymid-p1[i+3]);

}

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

{

a[0][0]=p1[i];

a[0][1]=p1[i+1];

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

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

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

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

}

setcolor(15);

for (i=0; i<8;i=i+2)

{

line(xmid+pi[i],ymid-pi[i+1],xmid+pi[i+2],ymid-pi[i+3]);

}

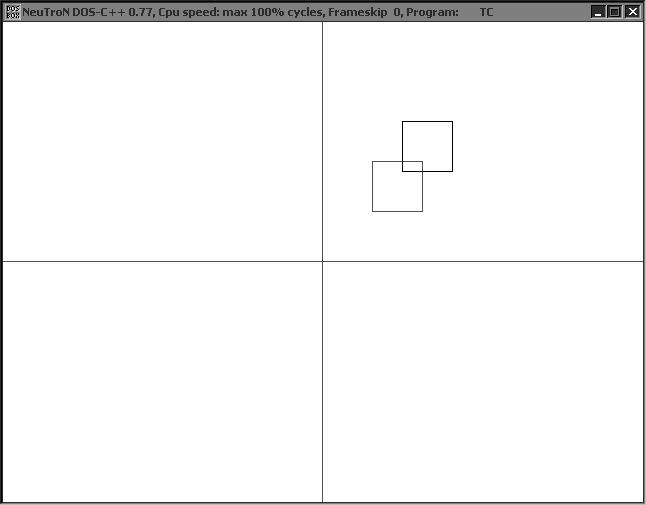
getch();

closegraph();

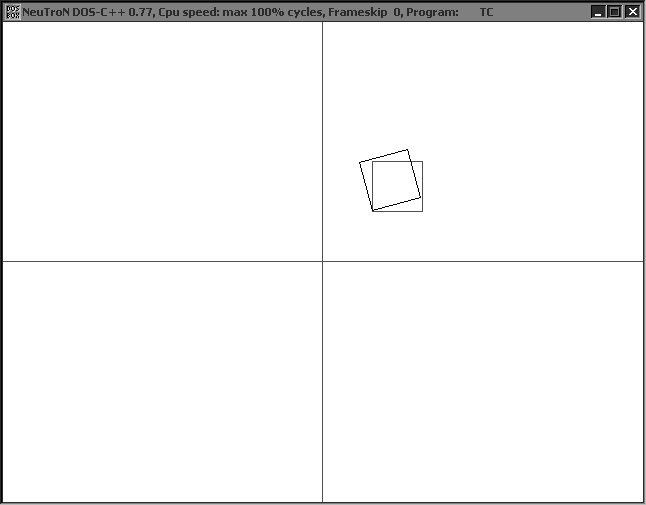
return 0;

}

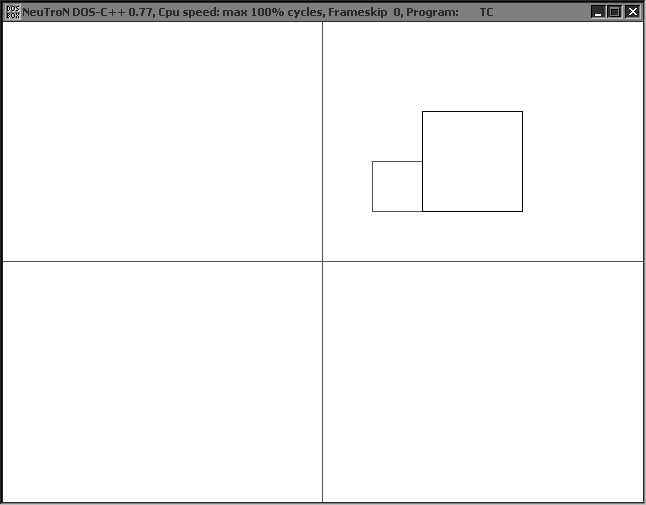
**Translation**

****

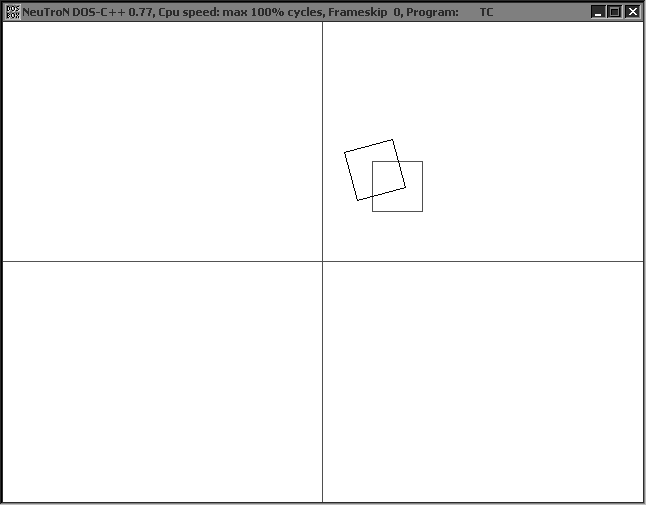
**Rotation**

****

**Scaling**

****

**Rotation about an arbitrary point**

****