

Practical 7

Task 1 : Write a program to perform Composite 2D-Transformation operations.

Source Code:

```
#include<graphics.h>
#include<stdio.h>
#include<math.h>

#define PI 3.14159265

int main()
{
    int gd = DETECT, gm;
    float xa, ya, xb, yb, xao, yao, xbo, ybo;

    printf("Enter the starting point\n");
    scanf("%f %f", &xa, &ya);
    printf("Enter the ending point\n");
    scanf("%f %f", &xb, &yb);
    xao=xa, yao=ya, xbo=xb, ybo=yb;

    int choice=0;
    while(choice!=6){
        choice=0;
        printf("Enter the type of transformation from list given\n");
        printf("1) Translation\n");
        printf("2) Rotation\n");
        printf("3) Scaling\n");
        printf("4) Shear X\n");
        printf("5) Shear Y\n");
        printf("6) Exit and print\n");
        scanf("%d", &choice);
        if (choice==1)
        {
            int ox, oy;
            printf("Enter new coordinates for Translation origin\n");
            scanf("%d %d", &ox, &oy);
            xa=xa+ox;
            xb=xb+ox;
            ya=ya+oy;
```

```

    yb=yb+oy;
}
else if(choice==2)
{
    int rx,ry;
    double ang,val;
    printf("Enter coordinates for point about which should i rotate\n");
    scanf("%d %d",&rx,&ry);
    printf("Enter angle by which to rotate\n");
    scanf("%lf",&ang);
    xa=xa+rx;
    xb=xb+rx;
    ya=ya+ry;
    yb=yb+ry;

    val = PI / 180.0;
    ang=ang*val;
    float nxa=xa,nya=ya,nxb=xb;

    xa = (nxa*cos(ang))-(nya*sin(ang));
    ya = (nxa*sin(ang))+(nya*cos(ang));
    xb = (nxb*cos(ang))-(nyb*sin(ang));
    yb = (nxb*sin(ang))+(nyb*cos(ang));
    xa=xa-rx;
    xb=xb-rx;
    ya=ya-ry;
    yb=yb-ry;

}
else if (choice==3)
{
    int sx,sy;
    printf("Enter scaling factors for x and y directions\n");
    scanf("%d %d",&sx,&sy);
    int rx,ry;
    printf("Enter coordinates for point about which should i Scale\n");
    scanf("%d %d",&rx,&ry);
    xa=xa+rx;
    xb=xb+rx;
    ya=ya+ry;
    yb=yb+ry;

    xa=xa*sx;
    xb=xb*sx;
    ya=ya*sy;

```

```

    yb=yb*sy;

    xa=xa-rx;
    xb=xb-rx;
    ya=ya-ry;
    yb=yb-ry;
}
else if (choice==4)
{
    int sx;
    printf("Enter Shear factors for x direction\n");
    scanf("%d",&sx);
    xa=xa+sx*ya;
    xb=xb+sx*yb;

}
else if(choice==5)
{
    int sy;
    printf("Enter Shear factors for y direction\n");
    scanf("%d",&sy);
    ya=ya+sy*xa;
    yb=yb+sy*xb;
}

}

initgraph(&gd,&gm,NULL);
line(xa,ya,xb,yb);
setlinestyle(DASHED_LINE,0,THICK_WIDTH);
line(xao,yao,xbo,ybo);
delay(5000);
closegraph();
return 0;
}

```

Output:

```
adnrs96@aditya-hp-envy-15-notebook-pc:/media/adnrs96/Local Disk/Local Disk(G)/CG
$ gcc prac_7_2DComposite_transformation_algo.c -lgraph -lm
adnrs96@aditya-hp-envy-15-notebook-pc:/media/adnrs96/Local Disk/Local Disk(G)/CG
$ ./a.out
Enter the starting point
10
10
Enter the ending point
100
100
Enter the type of transformation from list given
1) Translation
2) Rotation
3) Scaling
4) Shear X
5) Shear Y
6) Exit and print
1
Enter new coordinates for Translation origin
60
70
Enter the type of transformation from list given
1) Translation
2) Rotation
3) Scaling
4) Shear X
5) Shear Y
6) Exit and print
2
Enter coordinates for point about which should i rotate
1
2
Enter angle by which to rotate
-10
Enter the type of transformation from list given
1) Translation
2) Rotation
3) Scaling
4) Shear X
5) Shear Y
6) Exit and print
6
[xch] Unknown sequence number while processing queue
```



DASHED_LINE is the original line.