Downloaded from www.arunupadhayay.com.np

2 – DIMENSIONAL TRANSFORMATION

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
#include<math.h>
#include<stdlib.h>
void menu();
void input();
void output();
void translation();
void rotation();
void scaling();
void shearing();
void reflection();
int a[10][2],i,x,option,temp,angle,tx,ty,fx,fy,sh,k,n,axis,y;
float sx,sy;

void menu()
{
        printf("menu\n");
        printf("1.Translation\n");
        printf("2.rotation\n");
        printf("3.scaling\n");
        printf("4.shearing\n");
        printf("5.reflection\n");
        printf("6.exit\n");
        printf("enter the choice:");
        scanf("%d",&option);
        switch(option)
        {
                case  1:
                        input();
                        translation();
                        break;
                case 2:
                        input();
                        rotation();
                        break;
                case 3:
```

Downloaded from www.arunupadhayay.com.np

```
                        input();
                        scaling();
                        break;

                case 4 :
                        input();
                        shearing();
                        break;
                case 5:
                        input();
                        reflection();
                        break;
                case 6:
                        exit(0);
                        break;
        }
}

void input()
{
        printf("enter the number of vertices:" );
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("enter the coordinates:");
                scanf("%d%d%d%d",&a[i][0],&a[i][1],&a[i+1][0],&a[i+1][1]);
        }
}


void output()
{
        cleardevice();
        for(i=0;i<n;i++)
        {
                line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]);
        }
}
```

Downloaded from www.arunupadhayay.com.np

```
void translation()
{
        output();
        printf("enter the tranformation vertex tx,ty:\n");
        scanf("%d%d",&tx,&ty);
        for(i=0;i<=n;i++)
        {
                a[i][0]=a[i][0]+tx;
                a[i][1]=a[i][1]+ty;
        }
        output();
        delay(10);
        menu();
}

void rotation()
{
        output();
        printf("enter the rotating angle:");
        scanf("%d",&y);
        printf("enter the pivot point:");
        scanf("%d%d",&fx,&fy);
        k=(y*3.14)/180;
        for(i=0;i<=n;i++)
        {
                a[i][0]=fx+(a[i][0]-fx)*cos(k)-(a[i][1]-fy)*sin(k);
                a[i][1]=fy+(a[i][0]-fx)*sin(k)-(a[i][1]-fy)*cos(k);
        }
        output();
        delay(10);
        menu();
}
```

Downloaded from www.arunupadhayay.com.np

```
void scaling()
{
        output();
        printf("enter the scaling factor\n");
        scanf("%f%f",&sx,&sy);
        printf("enter the fixed point:");
        scanf("%d%d",&fx,&fy);
        for(i=0;i<=n;i++)
        {
                a[i][0]=a[i][0]*sx+fy*(1-sx);
                a[i][1]=a[i][1]*sy+fy*(1-sy);
        }
        output();
        delay(10);
        menu();
}

void shearing()
{
        output();
        printf("enter the shear value:");
        scanf("%d",&sh);
        printf("enter the fixed point:");
        scanf("%d%d",&fx,&fy);
        printf("enter the axis for shearing if x-axis then 1 if y-axis the 0:");
        scanf("%d",&axis);
        for(i=0;i<=n;i++)
        {
                if(axis==1)
                {
                        a[i][0]=a[i][0]+sh*(a[i][1]-fy);
                }
                else
                {
                        a[i][1]=a[i][1]+sh*(a[i][0]-fx);
                }
        }
        output();
        delay(10);
        menu();
}
```

Downloaded from www.arunupadhayay.com.np

```
void reflection()
{
        output();
        for(i=0;i<=n;i++)
        {
                temp=a[i][0];
                a[i][0]=a[i][1];
                a[i][1]=temp;
        }
        output();
        delay(10);
        menu();
}


void main()
{
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"c:\\tcplus\\bgi");
        menu();
        getch();
}
```

Downloaded from www.arunupadhayay.com.np

OUTPUT

Menu

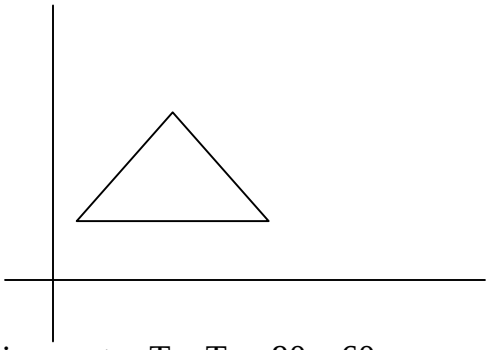
1. Translation
2. Rotation
3. Scaling
4. Shearing
5. Reflection
6. Exit

TRANSLATION

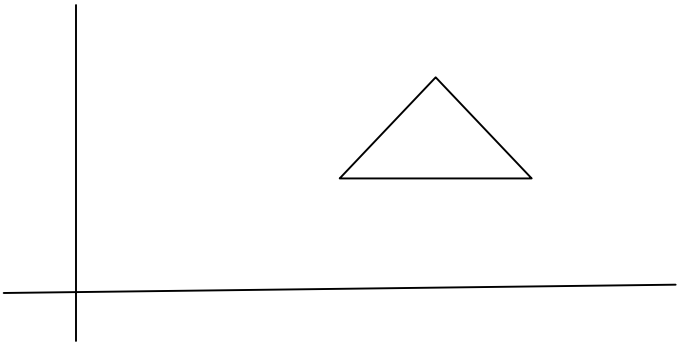
Enter the choice : 1

Enter the number of Vertices: 3

Enter the coordinates : 30 150 10 200
Enter the coordinates : 10 200 60 200
Enter the coordinates : 60 200 30 150



Enter the translation vector Tx, Ty : 90 60



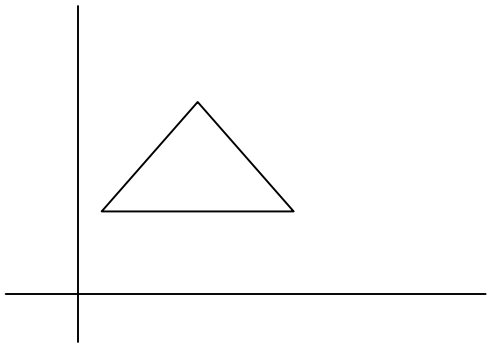
Downloaded from www.arunupadhayay.com.np

ROTATION

Enter the choice : 2

Enter the number of Vertices: 3

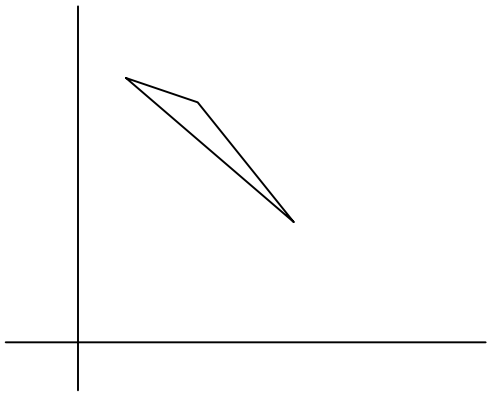
Enter the coordinates : 30 150 10 200
Enter the coordinates : 10 200 60 200
Enter the coordinates : 60 200 30 150



Enter the Rotating Angle : 90

Enter the Pivot Point : 100 200

`



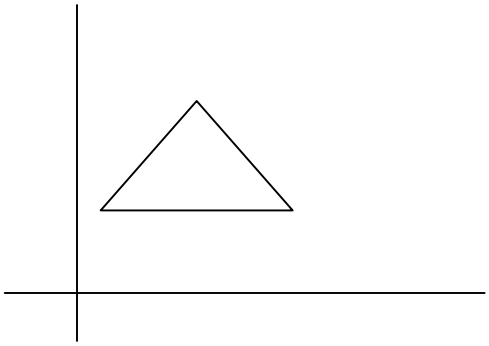
Downloaded from www.arunupadhayay.com.np

SCALING

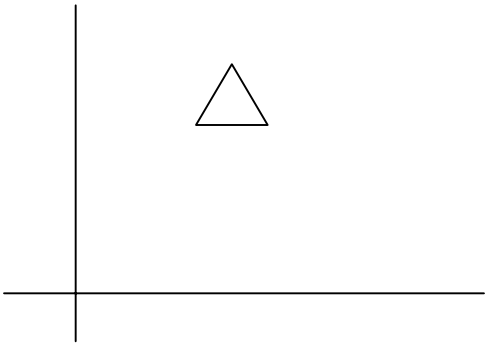
Enter the choice : 3

Enter the number of Vertices: 3

Enter the coordinates : 30 150 10 200
Enter the coordinates : 10 200 60 200
Enter the coordinates : 60 200 30 150



Enter the scaling Factor : 0.3 0.4
Enter the Fixed Point : 100 200



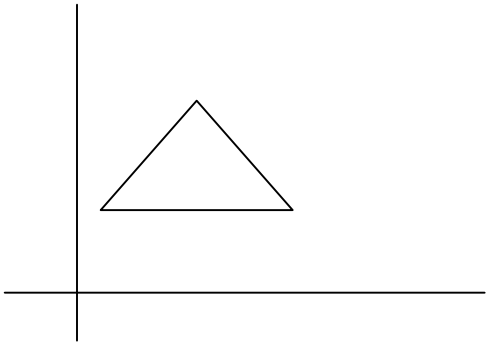
Downloaded from www.arunupadhayay.com.np

SHEARING

Enter the choice : 4

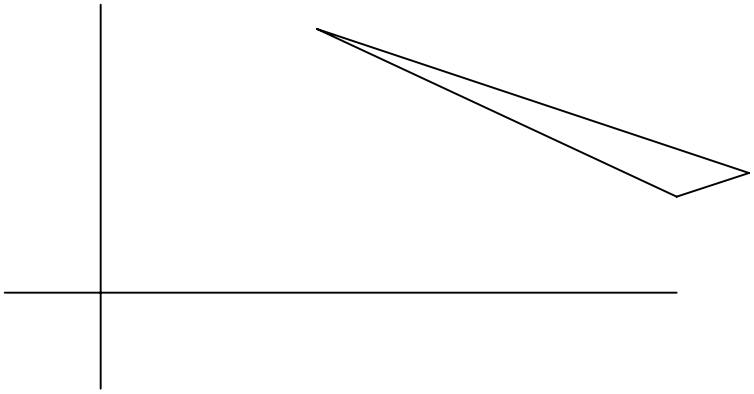
Enter the number of Vertices: 3

Enter the coordinates : 30 150 10 200
Enter the coordinates : 10 200 60 200
Enter the coordinates : 60 200 30 150



Enter the shear Value : 5
Enter the fixed point : 50 100

Enter the Axis for shearing if x-axis then 1
if y-axis then 0



Downloaded from www.arunupadhayay.com.np

REFLECTION

Enter the choice : 5

Enter the number of Vertices: 3

Enter the coordinates : 30 150 10 200
Enter the coordinates : 10 200 60 200
Enter the coordinates : 60 200 30 150