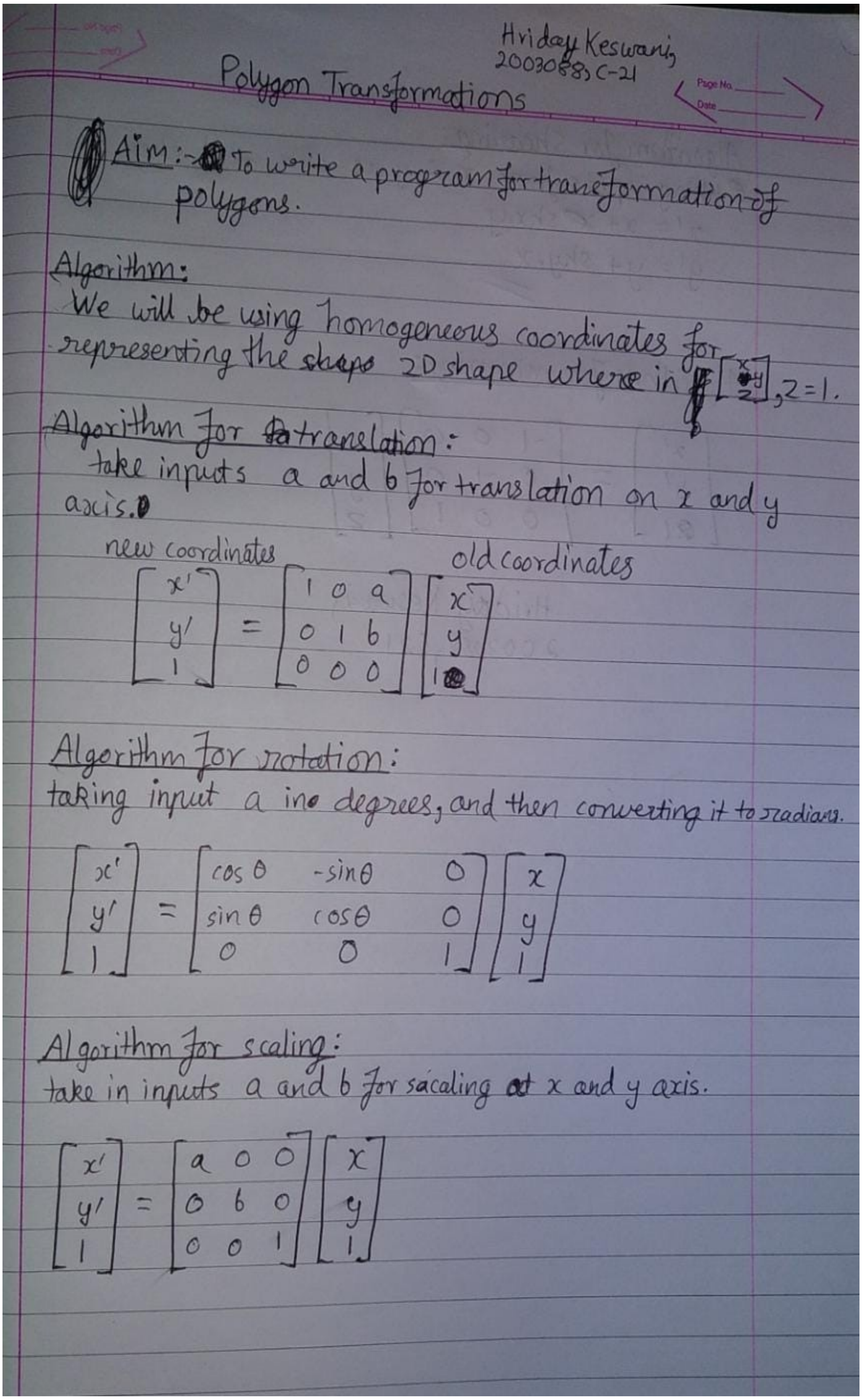
CG-Assignment
Transformations of 2D Polygons

Program:
Writing a program to Transform 2D polygons



Hriday Keswani,
2003088, C-21

Page No. ______
Date ______

Polygon Transformations

Aim:- To write a program for transformation of polygons.

Algorithm:
We will be using homogeneous coordinates for representing the ~~shape~~ 2D shape where in $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$, z=1.

Algorithm for ~~ta~~ translation:
take inputs a and b for translation on x and y axis.

new coordinates old coordinates

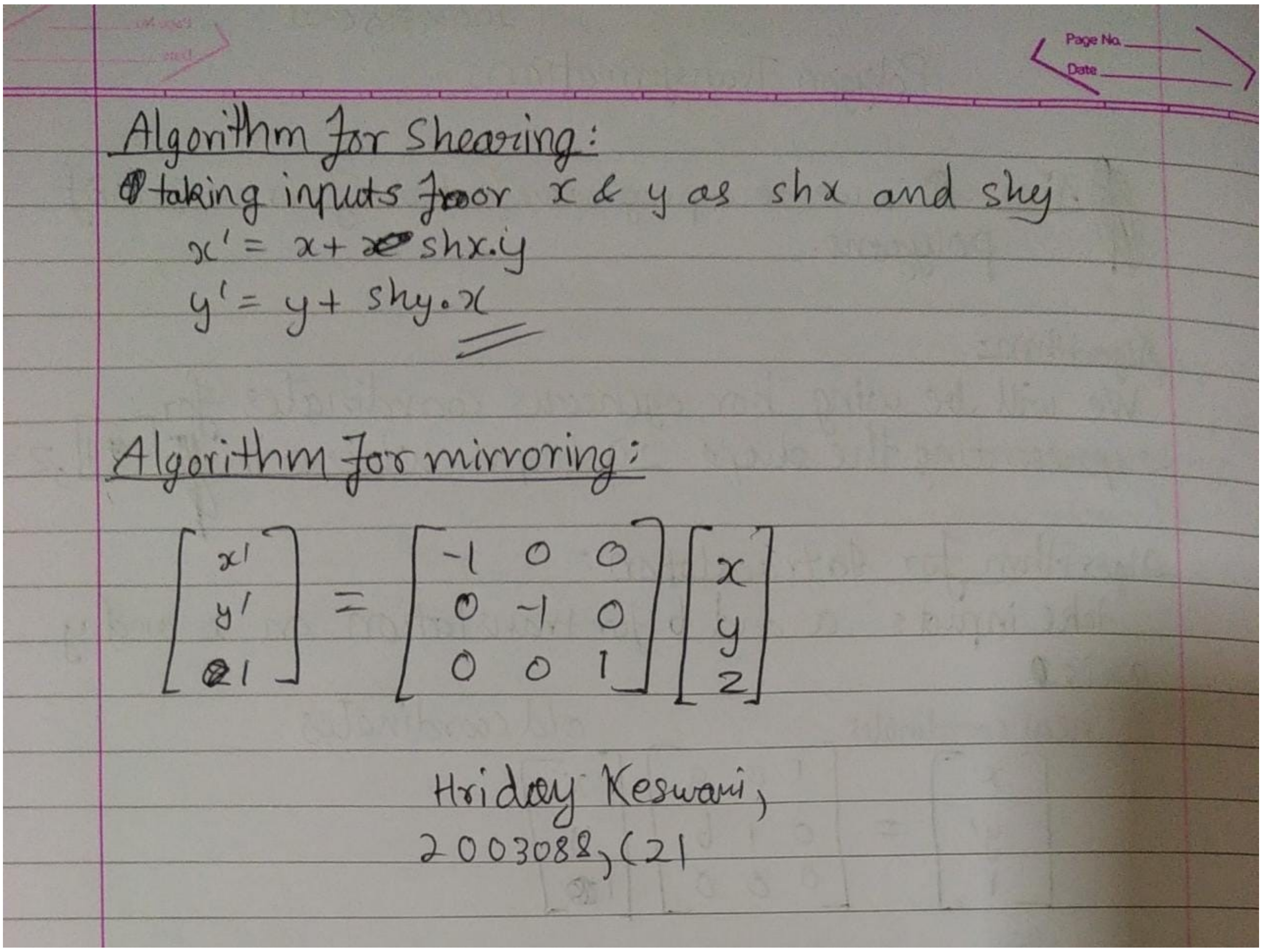
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Algorithm for rotation:
taking input a in degrees, and then converting it to radians.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Algorithm for scaling:
take in inputs a and b for sacaling ~~at~~ x and y axis.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Algorithm for Shearing:

taking inputs for x & y as shx and shy

$x' = x + shx.y$

$y' = y + shy.x$

Algorithm for mirroring:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Hriday Keswani,
2003088, C21

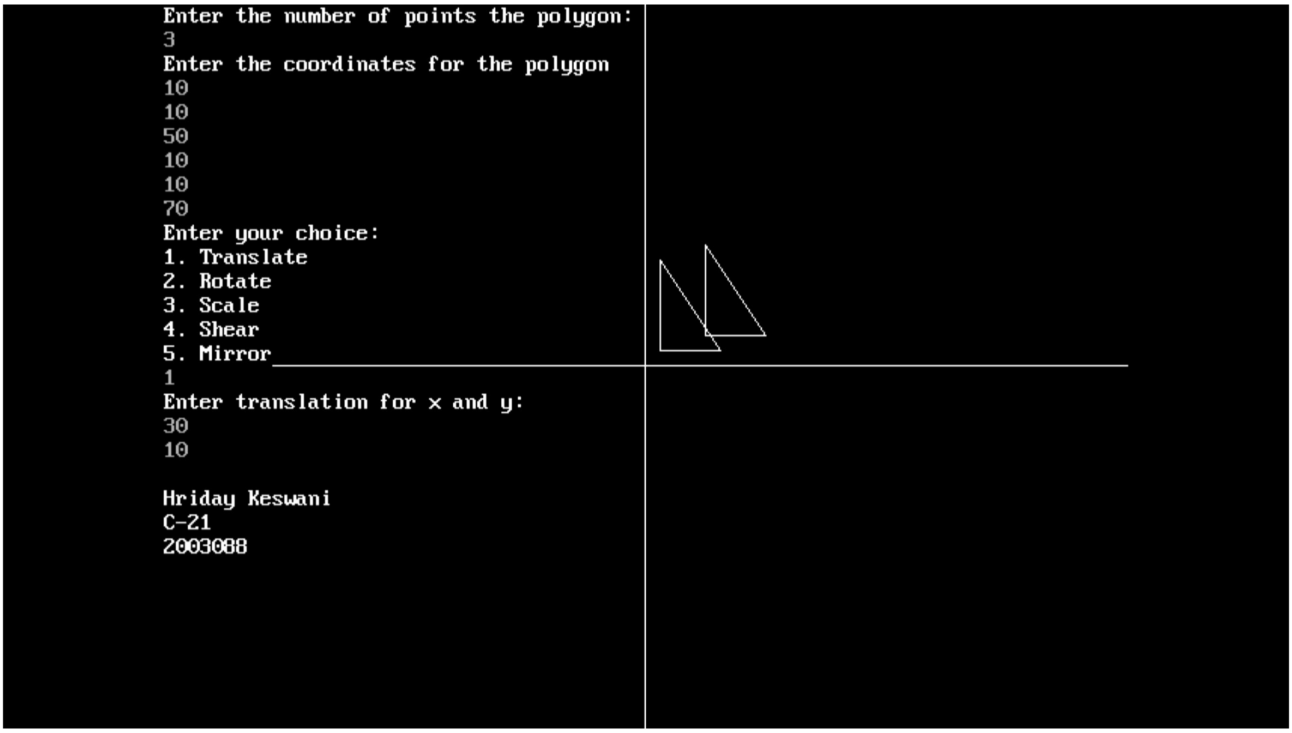
Code:

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

float shape[100][3];

void drawP(int pts, float arr[100][3]);
void addPoint(float x, float y, int index);
void translate(float tx, float ty, int pts, float arr[100][3]);
void rotate(float rot, int pts, float arr[100][3]);
void scale(float sx, float sy, int pts, float arr[100][3]);
void shear(float x, float y, int pts, float arr[100][3]);
void mirror(int pts, float arr[100][3]);

void main(){
    int gd = DETECT, gm;
    int n,i,choice;
    float x,y,a,b;

    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    line(getmaxx()/2,0,getmaxx()/2,getmaxy());
    line(0,getmaxy()/2,getmaxx(),getmaxy()/2);

    printf("Enter the number of points the polygon:\n");
    scanf("%d",&n);

    printf("Enter the coordinates for the polygon\n");
    for(i=0;i<n;i++){
     scanf("%f%f",&x,&y);
     addPoint(x,y,i);
     }

    drawP(n,shape);

    printf("Enter your choice:\n1. Translate\n2. Rotate\n3. Scale\
n4. Shear\n5. Mirror\n");
    scanf("%d",&choice);
    switch(choice){
     case 1:
     printf("Enter translation for x and y:\n");
     scanf("%f%f",&a,&b);
     translate(a,b,n,shape);
     break;
     case 2:
     printf("Enter rotaion angle:\n");
     scanf("%f",&a);
     rotate(a,n,shape);
     break;
     case 3:
     printf("Enter scale for x and y:\n");
     scanf("%f%f",&a,&b);
     scale(a,b,n,shape);
     break;
     case 4:
     printf("Enter shear for x and y:\n");
     scanf("%f%f",&a,&b);
     shear(a,b,n,shape);
     break;
     case 5:
     mirror(n,shape);
     break;
    }
    printf("\nHriday Keswani\nC-21\n2003088");
    getch();
    clrscr();
}

void translate(float tx, float ty, int pts, float arr[100][3]){
     int i;
```

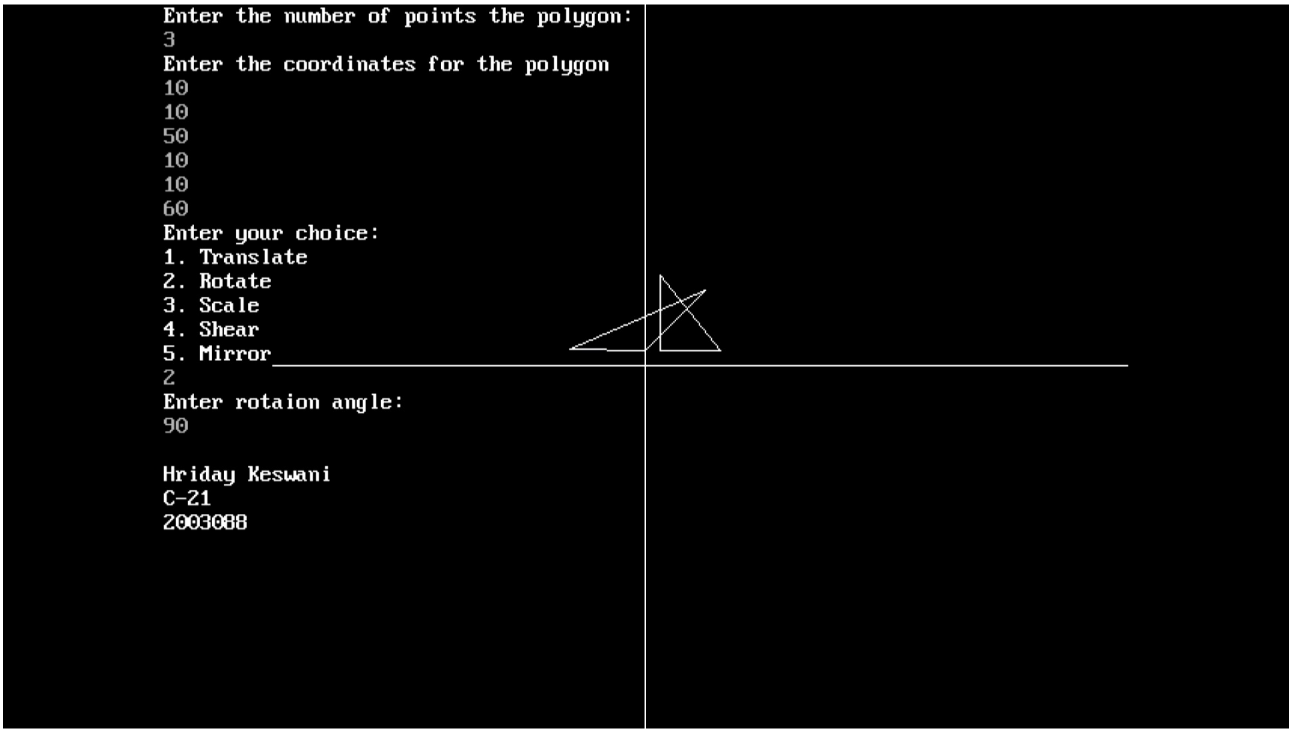
```
	for(i=0;i<pts;i++){
		arr[i][0]+=tx;
		arr[i][1]+=ty;
	}
	drawP(pts, shape);
}

void addPoint(float x, float y, int index){
	shape[index][0] = x;
	shape[index][1] = y;
	shape[index][2] = 1;
}

void drawP(int pts, float arr[100][3]){
	int i;
	for(i=0;i<pts-1;i++){
		line((getmaxx()/2)+arr[i][0],((getmaxy()/2)-arr[i][1]),
(getmaxx()/2)+arr[i+1][0],((getmaxy()/2)-arr[i+1][1]));
	}
	line((getmaxx()/2)+arr[0][0],((getmaxy()/2)-arr[0][1]),
(getmaxx()/2)+arr[pts-1][0],((getmaxy()/2)-arr[pts-1][1]));
}

void rotate(float rot, int pts, float arr[100][3]){
	int i;
	rot=rot*(3.142/180);
	for(i=0;i<pts;i++){
		arr[i][0]-=sin(rot)*arr[i][1];
		arr[i][1]+=sin(rot)*arr[i][0];
	}
	drawP(pts, arr);
}

void scale(float sx, float sy, int pts, float arr[100][3]){
	int i;
	for(i=0;i<pts;i++){
		arr[i][0]*=sx;
		arr[i][1]*=sy;
	}
	drawP(pts, arr);
}

void shear(float x, float y, int pts, float arr[100][3]){
	int i;
	for(i=0;i<pts;i++){
		arr[i][0]+=x*arr[i][1];
		arr[i][1]+=y*arr[i][0];
	}
	drawP(pts,arr);
}
```

```
void mirror(int pts, float arr[100][3]){
     int i;
     for(i=0;i<pts;i++){
          arr[i][0]*=-1;
          arr[i][1]*=-1;
     }
     drawP(pts,arr);
}
```

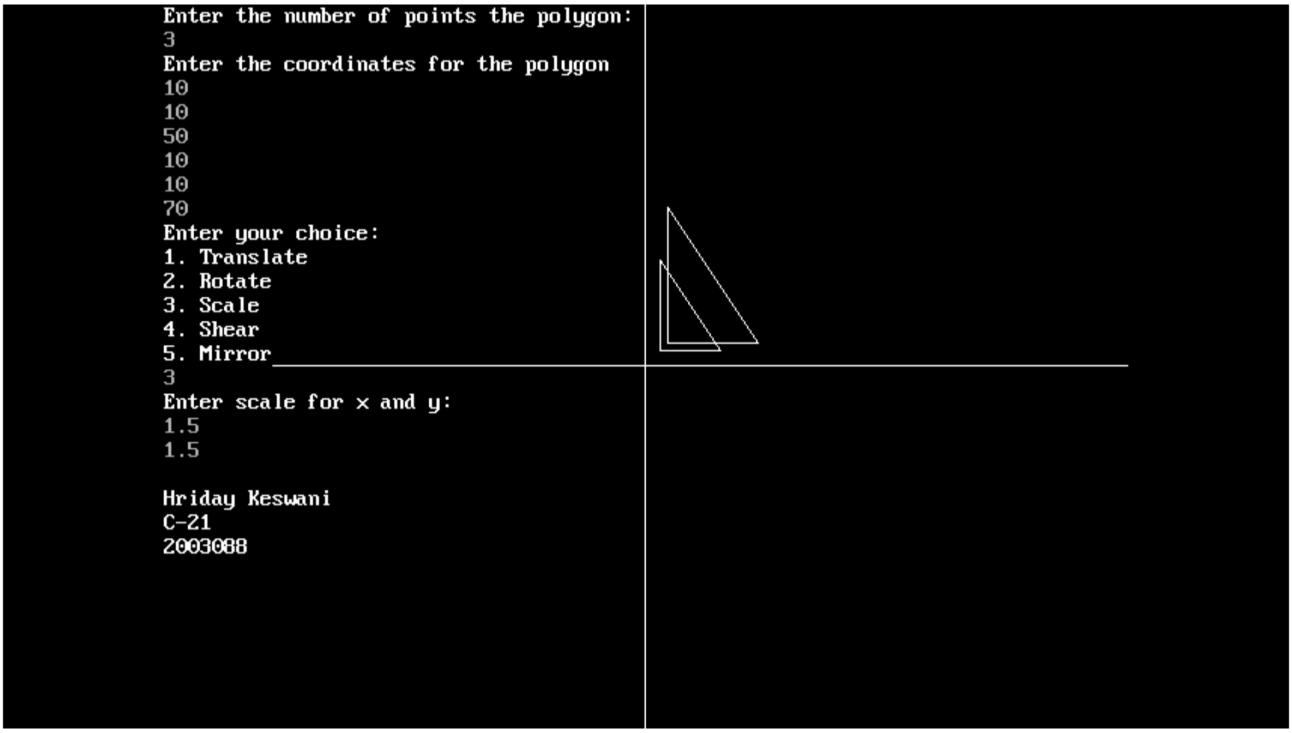
Output:



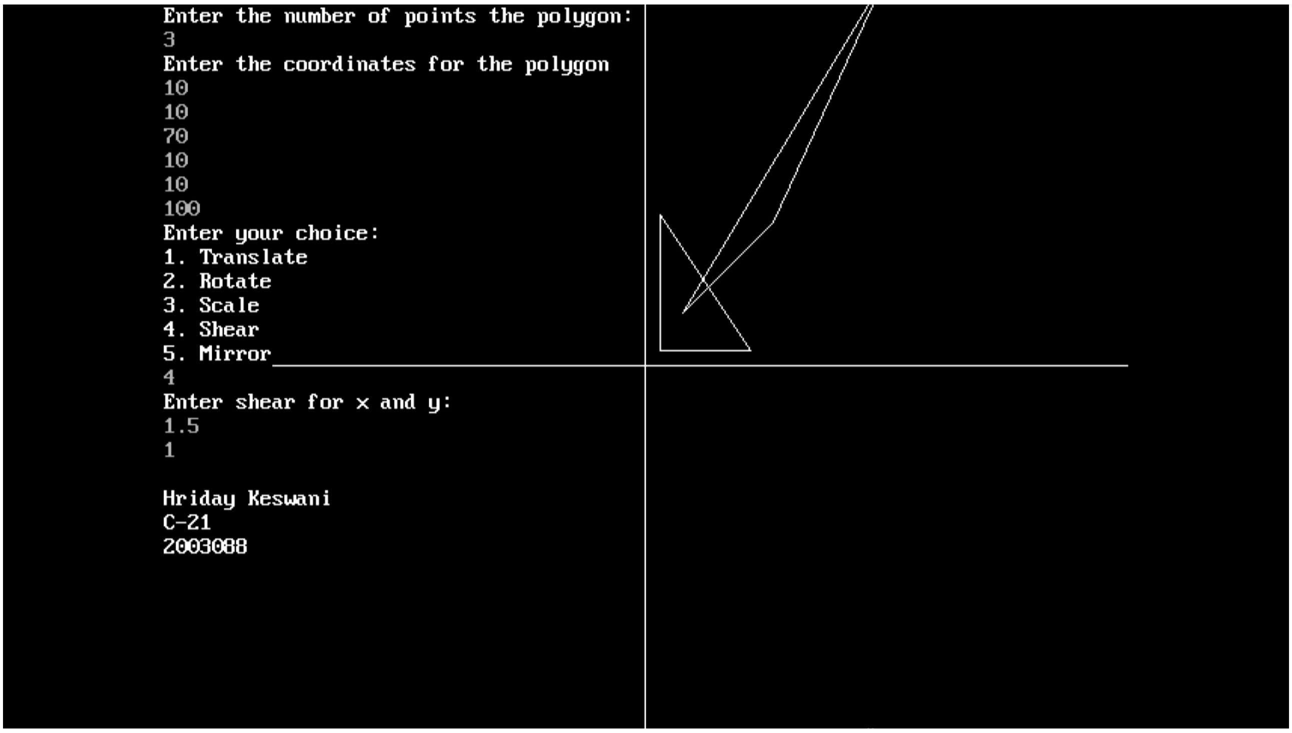
```
Enter the number of points the polygon:
3
Enter the coordinates for the polygon
10
10
50
10
10
70
Enter your choice:
1. Translate
2. Rotate
3. Scale
4. Shear
5. Mirror
1
Enter translation for x and y:
30
10

Hriday Keswani
C-21
2003088
```



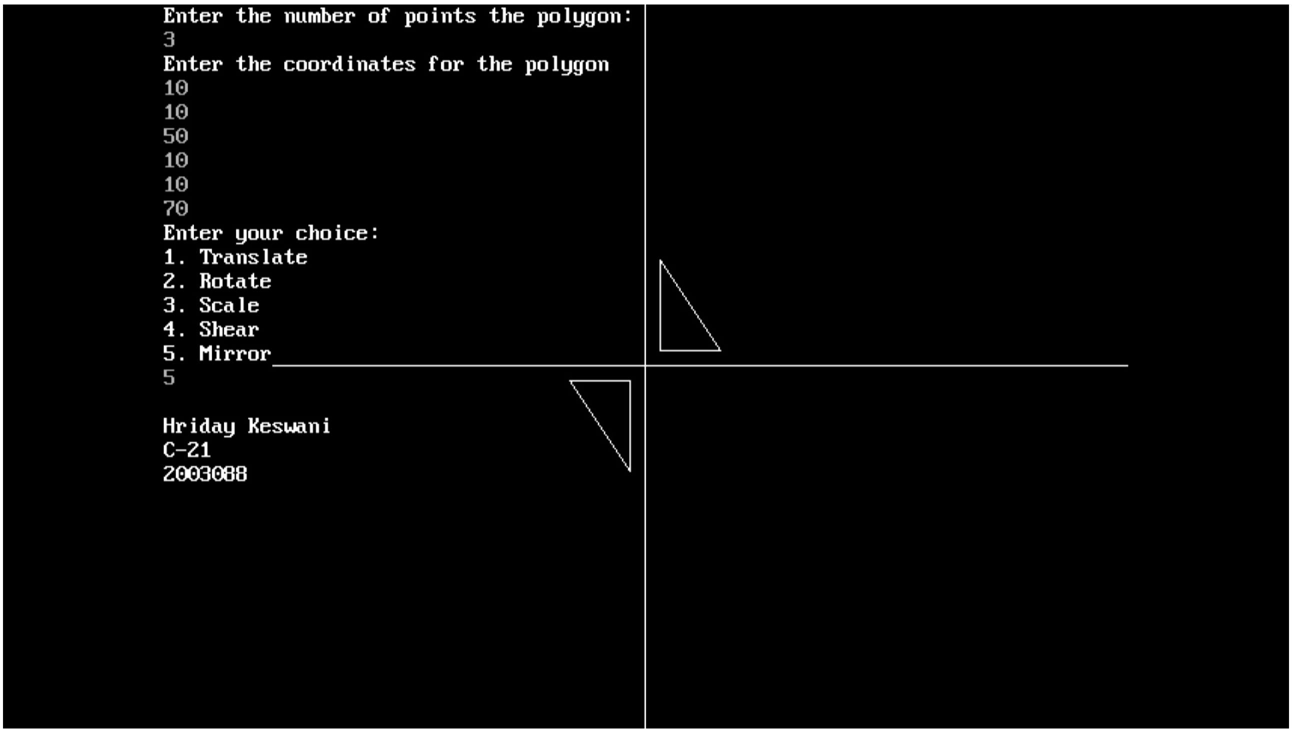
```
Enter the number of points the polygon:
3
Enter the coordinates for the polygon
10
10
50
10
10
60
Enter your choice:
1. Translate
2. Rotate
3. Scale
4. Shear
5. Mirror
2
Enter rotaion angle:
90

Hriday Keswani
C-21
2003088
```



```
Enter the number of points the polygon:
3
Enter the coordinates for the polygon
10
10
50
10
10
70
Enter your choice:
1. Translate
2. Rotate
3. Scale
4. Shear
5. Mirror
3
Enter scale for x and y:
1.5
1.5

Hriday Keswani
C-21
2003088
```



```
Enter the number of points the polygon:
3
Enter the coordinates for the polygon
10
10
70
10
10
100
Enter your choice:
1. Translate
2. Rotate
3. Scale
4. Shear
5. Mirror
4
Enter shear for x and y:
1.5
1

Hriday Keswani
C-21
2003088
```



```
Enter the number of points the polygon:
3
Enter the coordinates for the polygon
10
10
50
10
10
70
Enter your choice:
1. Translate
2. Rotate
3. Scale
4. Shear
5. Mirror
5

Hriday Keswani
C-21
2003088
```