

Code:

```
#include<bits/stdc++.h>
#include<graphics.h>
#include<math.h>
#include<stdio.h>

void draw(int x[],int y[],int n);
void translation(int x[],int y[],int n);
void scale(int x[],int y[],int n);
void shear(int x[],int y[],int n);
void rotation(int x[],int y[],int n);

using namespace std;

int main(){
    int Xmin,Ymin,Xmax,Ymax;
    initwindow(800,800);
    Xmin=20;
    Xmax=800-20;
    Ymin=50;
    Ymax=800-50;
    rectangle(Xmin,Ymin,Xmax,Ymax);
    line(Xmax/2,Ymin,Xmax/2,Ymax);
    line(Xmin,Ymax/2+80,Xmax,Ymax/2+80)
;
    outtextxy(450,20,"ID: 174049");

    int n, x[100], y[100], i;
    cout << "Enter no. of sides in polygon: ";
    cin >> n;
    cout << "Enter coordinates x, y for each
vertex: ";
    for (i = 0; i < n; i++) {
        cin >> x[i] >> y[i];
    }

    n=4;
    x[0]=550;x[1]=650;x[2]=650;x[3]=550;
    y[0]=300;y[1]=300;y[2]=200;y[3]=200;
    draw(x,y,n);
    int choice;
    cout<<"1. Translation"<<endl;
    cout<<"2. Scale"<<endl;
    cout<<"3. Shear"<<endl;
    cout<<"4. Rotation"<<endl;
    cout<<"5. Exit"<<endl;
    while (1){
```

```
        cout<<"Enter your choice: ";
        cin>>choice;
        switch(choice){
            case 1:
                translation(x,y,n);
                break;
            case 2:
                scale(x,y,n);
                break;
            case 3:
                shear(x,y,n);
                break;
            case 4:
                rotation(x,y,n);
                break;
            case 5:
                exit(0);
        }
    }

    while (!kbhit()){
        delay(100);
    }

}

void draw(int x[100],int y[100],int n)
{
    for (int i = 0; i < n; i++) {
        setcolor(i+10);
        line(x[i], y[i], x[(i + 1) % n], y[(i +
1) % n]);
    }
}

void initialization(int x[],int xt[],int n){
    for (int i=0;i<n;i++){
        x[i]=xt[i];
    }
}

void translation(int xt[],int yt[],int n){
    int x[100], y[100];
    int tx,ty;
    cout<<"Enter tx,ty for translation: ";
    cin>>tx>>ty;
    for(int i=0;i<n;i++){
        x[i]=x[i]+tx;
        y[i]=y[i]+ty;
```

```

    }
    draw(x,y,n);
}
void scale(int xt[],int yt[],int n)
{
    int x[100], y[100];
    float sf;
    cout << "Enter scale factors: sf : ";
    cin >> sf;
    for (int i = 0; i < n; i++) {
        x[i] = x[0] + (int)((float)(x[i] -
x[0]) * sf);
        y[i] = y[0] + (int)((float)(y[i] -
y[0]) * sf);
    }
    setcolor(YELLOW);
    draw(x,y,n);
}
void shear(int xt[],int yt[],int n){
    int x[100], y[100];
    float shx,shy;
    char ch;
    delay(10);
    cout<<"enter the direction of shear : ";
    cin>>ch;
    if(ch=='x')
    {
        cout<<"enter xf-direction of
shear : ";
        cin>>shx;
        y[2]=y[2]+shx*x[2];
        y[1]=y[1]+shx*x[1];
        setcolor(RED);
        draw(x,y,n);
    }
    else
    {
        cout<<"enter y-direction of
shear : ";
        cin>>shy;
        x[2]=x[2]+shy*y[2];
        x[3]=x[3]+shy*y[3];
        setcolor(RED);
        draw(x,y,n);
    }
}

```

```

}
void rotation(int xt[],int yt[],int n){
    int x[100], y[100];
    float theta;
    printf("\nEnter the angle for rotation:
");
    scanf("%f",&theta);
    theta=theta*(3.14/180);

    int nx[100],ny[100];
    for (int i=0;i<n;i++){
        nx[i]=refx+(x[i]-refx)*cos(theta)-
(y[i]-refy)*sin(theta);
        ny[i]=refy+(x[i]-
refx)*sin(theta)+(y[i]-refy)*cos(theta);
    }
    setcolor(RED);
    draw(nx,ny,n);
}

```

Output:

