**//Plotting the coordinate Axis and Drawing figures with reference to it**

**#include<stdio.h>**

**#include<graphics.h>**

**#include<conio.h>**

**#include<math.h>**

*//some required global declaration*

**int n, xs[10][2] , j , i , tx , ty;**

**float x , y , a;**

**void draw();**

**void translate();**

**void Translation();**

**void Rotation();**

**void Scaling();**

**void Reflection();**

**int main()**

{

int inp;

int gd=DETECT,gm;

initgraph(&gd,&gm,"");

//Coordinate lines

line(320,0,320,480);

line(0,240,640,240);

setviewport(320,0,640,480,1); *//TO GET THE DIAGRAMS IN THE FIRST QUADRANT OF THE COORDINATE AXISES*

printf("\nEnter the number of Coordinates points of the polygon(Line = 2): ");

scanf("%d",&n); *//SIDES OF THE POLYGON MAX = 10*

printf("\nWhat you want to do with it.");

printf("\nRotation :: 1 \nTranslation :: 2 \nResize :: 3 \nReflection :: 4 \nInput :: ");

scanf("%d",&inp);

switch(inp)

{

case 1:

Rotation();

break;

case 2:

Translation();

break;

case 3:

Scaling();

break;

case 4:

Reflection();

break;

}

getch();

cleardevice();

}

/////////////////////////////////////////////////////////

**void Translation()**

**{**

printf("\nEnter coordinates x,y for each vertex: ");

for(i=0;i<n;i++)

{

scanf("%d%d",&xs[i][0],&xs[i][1]);

}

printf("\nEnter distance for translation tx and ty: ");

scanf("%d%d",&tx,&ty);

setcolor(RED);

draw();

translate();

setcolor(DARKGRAY);

draw();

**}**

**void translate()**

**{**

for(i=0;i<n;i++)

{

xs[i][0]+=tx;

xs[i][1]+=ty;

}

**}**

////////////////////////////////////////////////////////////

**void draw()**

**{**

for(i=0;i<n;i++)

line(xs[i][0],xs[i][1],xs[(i+1)%n][0],xs[(i+1)%n][1]);

**}**

/////////////////////////////////////////////////////////////

**void Rotation()**

**{**

printf("\nEnter coordinates x, y for each vertex: ");

for(i=0;i<n;i++)

{

scanf("%d%d",&xs[i][0],&xs[i][1]);

}

draw();

printf("\nEnter rotational angle: ");

scanf("%f",&a);

a=(3.14/180)\*a;

for(i=0;i<n;i++)

{

x=xs[i][0];

y=xs[i][1];

xs[i][0]=floor(x\*cos(a)-y\*sin(a));

xs[i][1]=floor(x\*sin(a)+y\*cos(a));

}

setcolor(RED);

draw();

**}**

/////////////////////////////////////////////////////////

**void Scaling()**

**{**

printf("\nEnter coordinates x,y for each vertex: ");

for(i=0;i< n;i++)

{

scanf("%d%d",&xs[i][0],&xs[i][1]);

}

printf("\nEnter the scaling vector: \n");

scanf("%d %d",&tx,&ty);

setcolor(WHITE);

draw();

for(i=0 ; i < n ; i++)

for(j = 0; j < 2 ; j++)

{

if(j == 0)

{

xs[i][j] = xs[i][j] \* tx;

}

else

{

xs[i][j] = xs[i][j] \* ty;

}

}

setcolor(RED);

draw();

**}**

//////////////////////////////////////////////////////////

**void Reflection()**

**{**

printf("\nEnter coordinates x,y for each vertex: ");

for(i=0;i< n;i++)

{

scanf("%d%d",&xs[i][0],&xs[i][1]);

}

setcolor(WHITE);

draw();

setviewport(0,0,320,480,1); *// TO GET THE REFLECTION IN 2ND QUADRANT*

setcolor(WHITE);

for(i=0;i<n;i++) *//MODIFIED DRAW() FUNCTION SHIFTING X - AXIS*

line(320 - xs[i][0], xs[i][1],320 - xs[(i+1)%n][0],xs[(i+1)%n][1]);

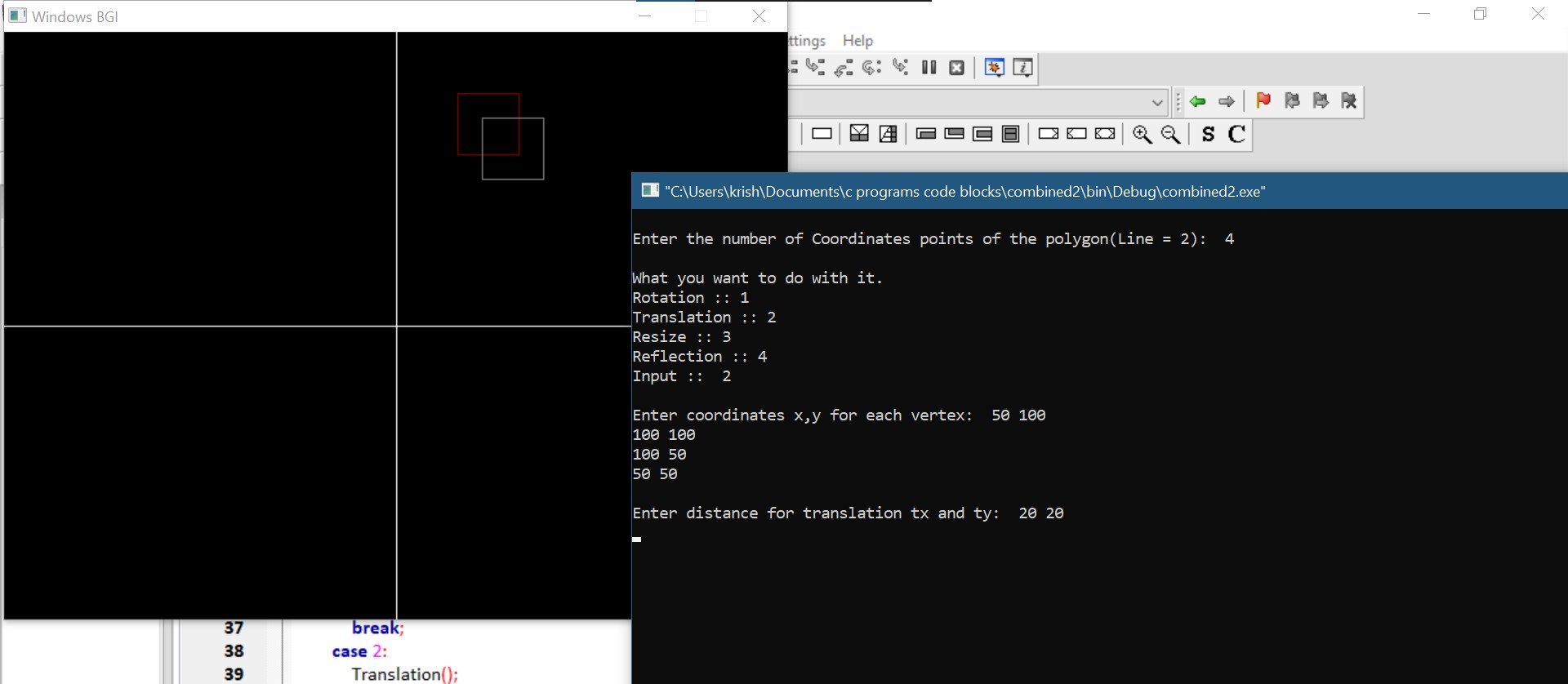
*//TO MAKE IT LOOK LIKE MIRROR*

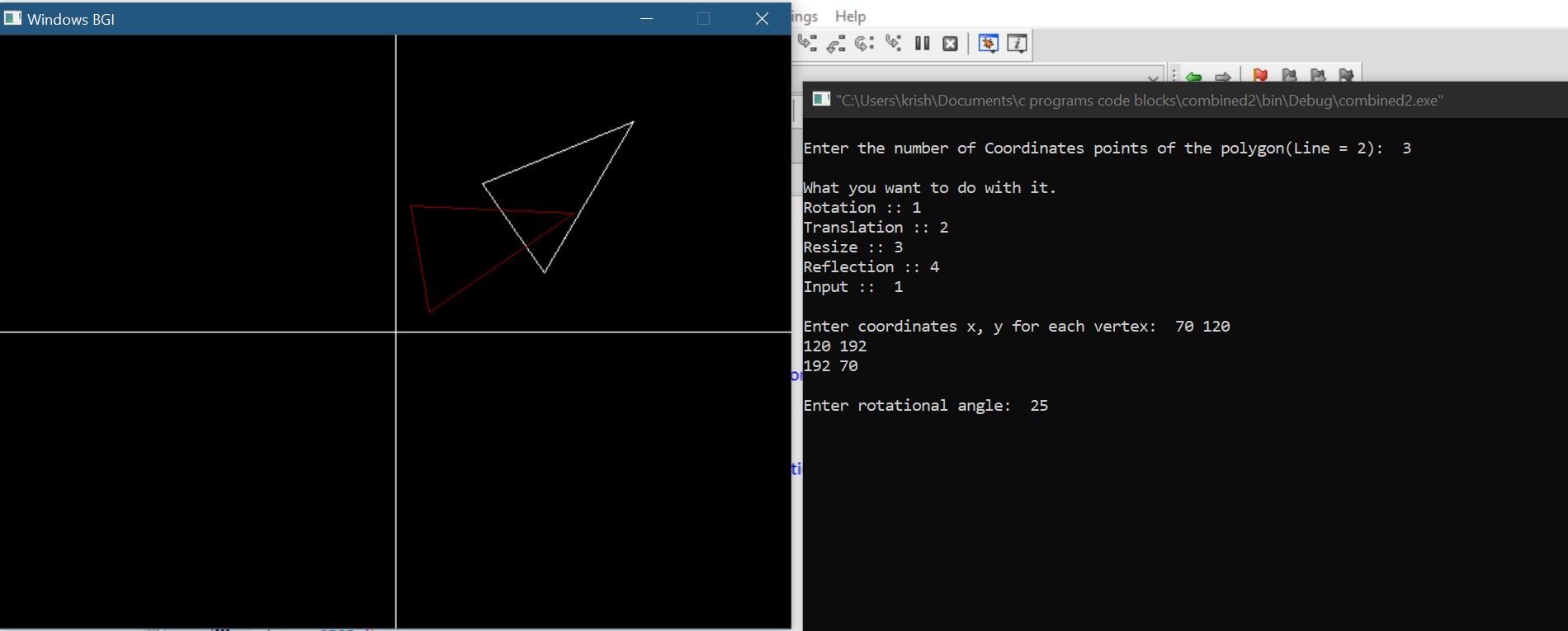
setcolor(WHITE);

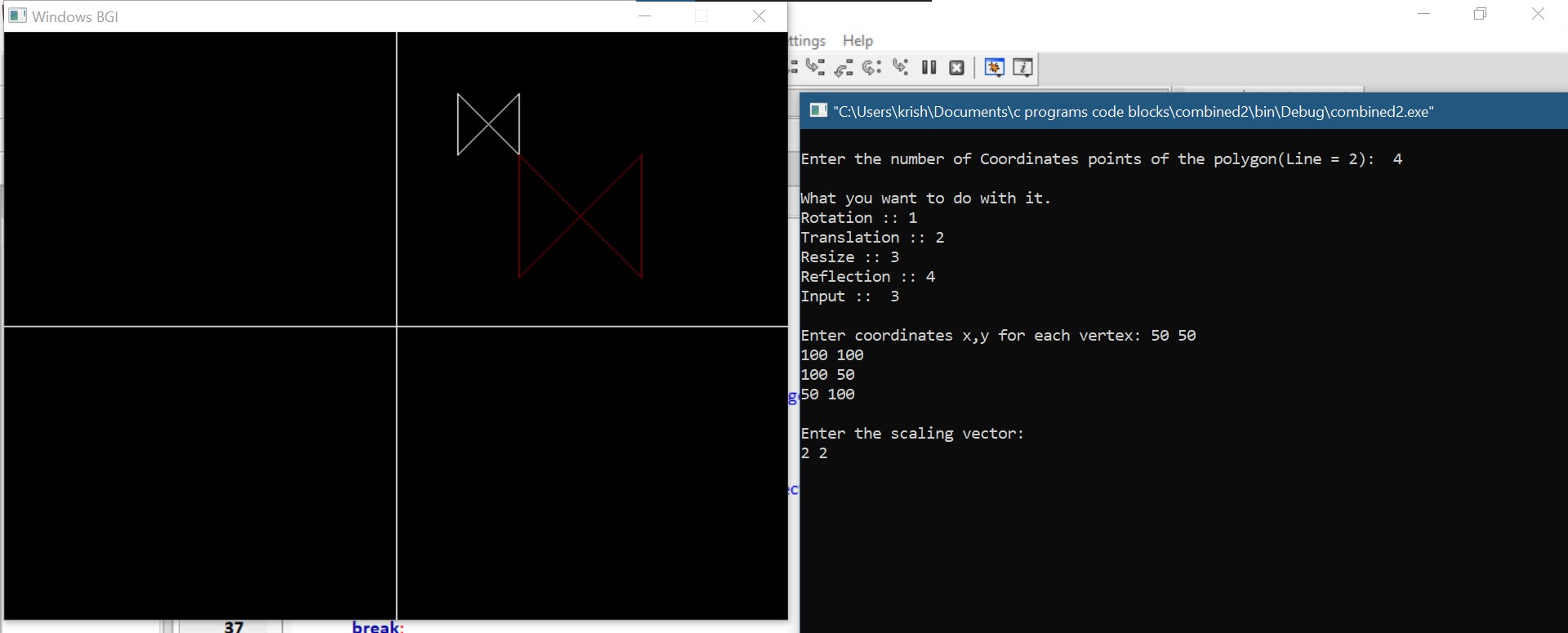
setlinestyle(0,4,1);

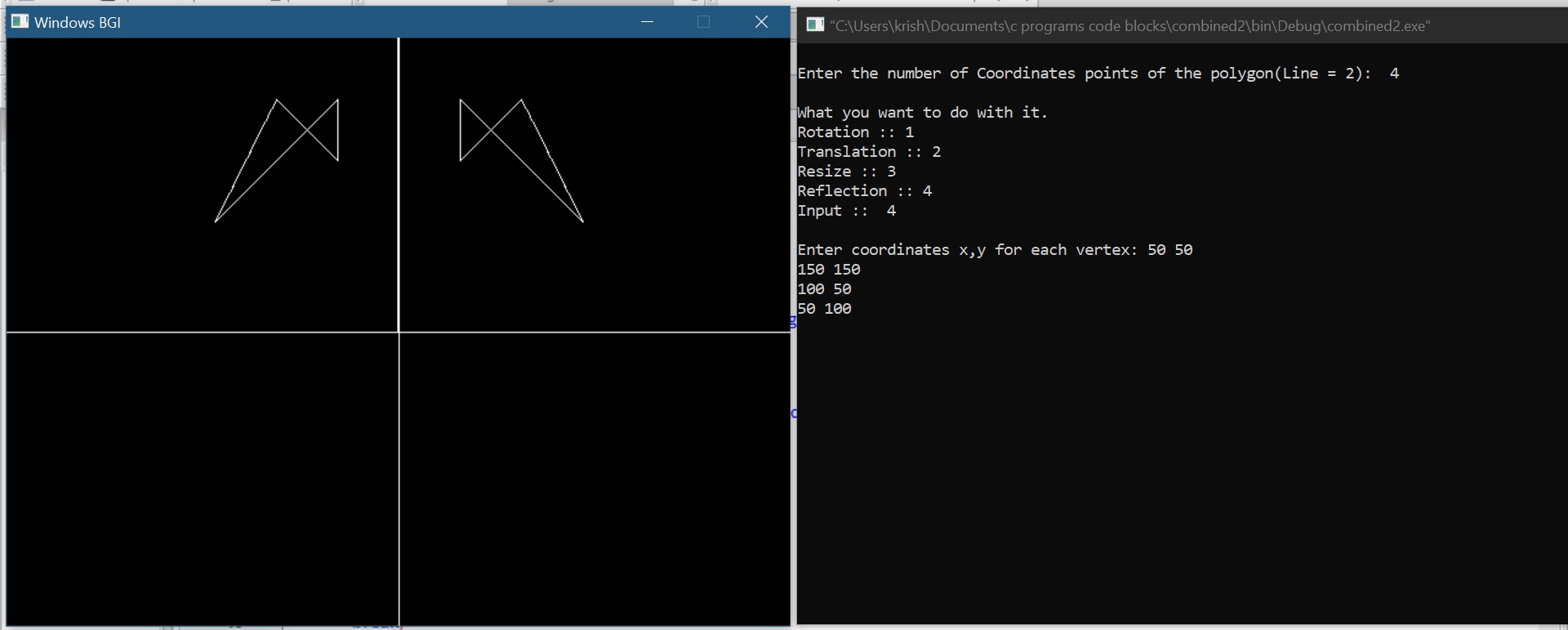
line(319,0,319,240);

**}**

****

****

****

****