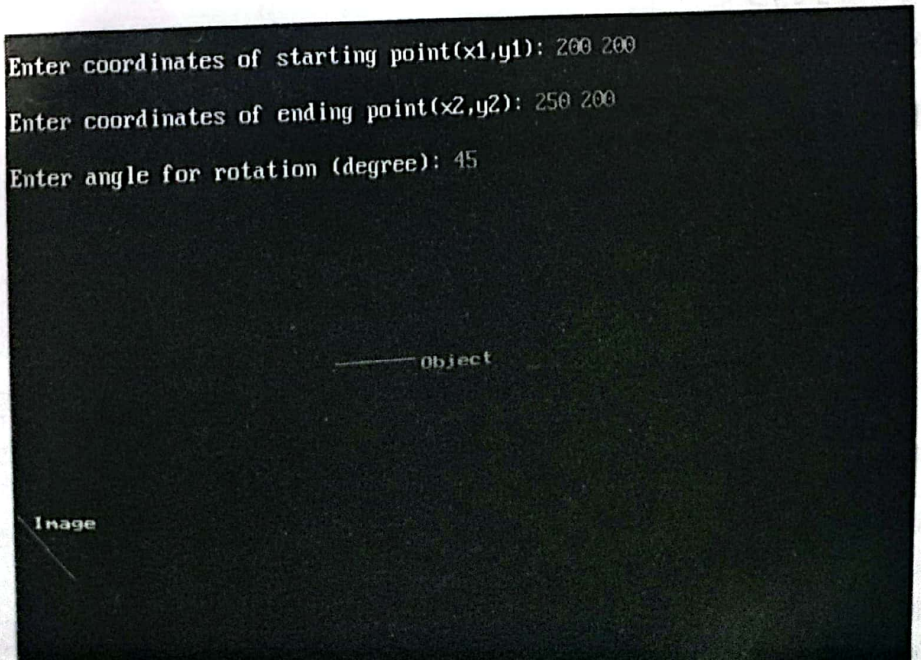
Lab problem 4.2: Program for Rotation

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
        int gd=DETECT,gm;
        float x1,y1,x2,y2,x3,y3,x4,y4,a,t;
        initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
        printf("\nEnter coordinates of starting point(x1,y1): ");
        scanf("%f%f",&x1,&y1);
        printf("\nEnter coordinates of ending point(x2,y2): ");
        scanf("%f%f",&x2,&y2);
        printf("\nEnter angle for rotation (degree): ");
        scanf("%f",&a);
        setcolor(7);
        line(x1,y1,x2,y2);
        outtextxy(x2+5,y2,"Object");
                t=a*(3.14/180);
                x3=(x1*cos(t))-(y1*sin(t));
                y3=(x1*sin(t))+(y1*cos(t));
                x4=(x2*cos(t))-(y2*sin(t));
                y4=(x2*sin(t))+(y2*cos(t));
        setcolor(15);
        line(x3,y3,x4,y4);
        outtextxy(x3+10,y3,"Image");
        getch();
}
```

Output:



$\begin{bmatrix} 1 \end{bmatrix}$ $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 \end{bmatrix}$ y, 1). Th

Example 4.2: Translate the given points (2, 5) by the translating value (3, 3)

Solution:

$(x, y) = (2, 5)$

$T_x = 3,$

$T_y = 3$

$x' = x + t_x$

$Y' = y + t_y$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \end{bmatrix}$$

Lab problem 4.1: Program for translation

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
        int gd=DETECT, gm;
        int x1,y1,x2,y2,tx,ty,x3,y3,x4,y4;
```

```
    initgraph(&gd, &gm,"C:\\TurboC3\\BGI");
    printf("\nEnter the starting point of line segment(x1,y1):");
    scanf("%d %d",&x1,&y1);
    printf("\nEnter the ending point of line segment(x2,y2):");
    scanf("%d %d",&x2,&y2);
    printf("\nEnter translation vector (tx,ty):");
    scanf("%d%d",&tx,&ty);
    setcolor(7);
    line(x1,y1,x2,y2);
    outtextxy(x2+5,y2,"Object");
        x3=x1+tx;
        y3=y1+ty;
        x4=x2+tx;
        y4=y2+ty;
  setcolor(15);
  line(x3,y3,x4,y4);
  outtextxy(x4+5,y4,"Image");
  getch();
}
```

Output

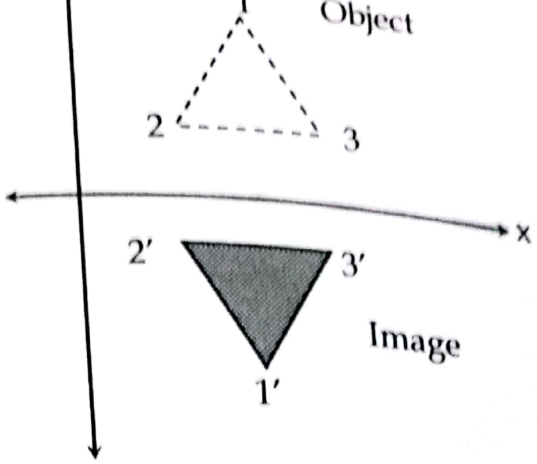


Figure 4.7: Reflection on x- axis

Lab problem 4.4: Program for reflection along x-axis

```
# include <stdio.h>
# include <conio.h>
# include <graphics.h>
# include <math.h>
char IncFlag;
int PolygonPoints[3][2] ={{10,100},{110,100},{110,200}};
void PolyLine()
{
        int iCnt;
        cleardevice();
        line(0,240,640,240);
        line(320,0,320,480);
        for (iCnt=0; iCnt<3; iCnt++)
        {
                line(PolygonPoints[iCnt][0],PolygonPoints[iCnt][1],
                PolygonPoints[(iCnt+1)%3][0],PolygonPoints[(iCnt+1)%3][1]);
        }
}
void Reflect()
{
```

```
        float Angle;
        int iCnt;
        int Tx,Ty;
        printf("endl");;
        for (iCnt=0; iCnt<3; iCnt++)
                PolygonPoints[iCnt][1] = (480 - PolygonPoints[iCnt][1]);
}
void main()
{
        int gDriver = DETECT, gMode;
        int iCnt;
        initgraph(&gDriver, &gMode, "C:\\TurboC3\\BGI");
        for (iCnt=0; iCnt<3; iCnt++)
        {
                PolygonPoints[iCnt][0] += 320;
                PolygonPoints[iCnt][1] = 240 - PolygonPoints[iCnt][1];
        }
        PolyLine();
        getch();
        Reflect();
        PolyLine();
        getch();
}
```

Output