

LAB :- 1

//DDA line drawing algorithm

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<graphics.h>
```

```
#include<math.h>
```

```
void main()
```

```
{
```

```
    int gd=DETECT,gm;
```

```
    int x1,y1,x2,y2,stepsize,dx,dy,i;
```

```
    float x,y,xinc,yinc;
```

```
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
```

```
    printf("Digital Differential Line Drawing Algorithm\n");
```

```
    printf("put the values of x1 and y1\n");
```

```
    scanf("%d%d",&x1,&y1);
```

```
    printf("put the values of x2 and y2\n");
```

```
    scanf("%d%d",&x2,&y2);
```

```
    dx=x2-x1;
```

```
    dy=y2-y1;
```

```
    x=x1;
```

```
    y=y1;
```

```
    if(abs(dy)>abs(dx))
```

```
    {
```

```
        stepsize=abs(dy);
```

```
    }
```

```
    else{
```

```
        stepsize=abs(dx);
```

```
    }
```

```
    xinc=dx/(float)stepsize;
```

```
    yinc=dy/(float)stepsize;
```

```
    putpixel(x,y,RED);
```

```
    for(i=0;i<stepsize;i++)
```

```
    {
```

```
        x=x+xinc;
```

```
        y=y+yinc;
```

```
        putpixel((int)(x+0.5),(int)(y+0.5),RED);
```


```
    }
```

```
    getch();
```

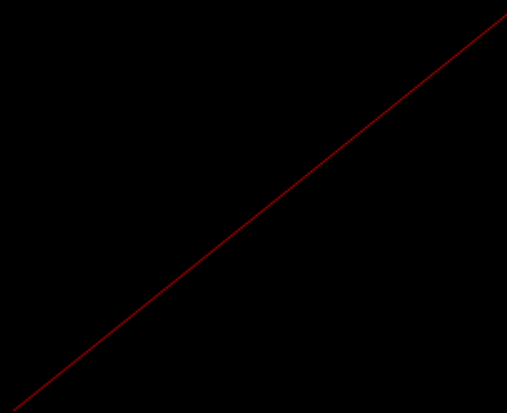
```
closegraph();  
}
```

OUTPUT OF LAB 1:-

```
Digital Differential Line Drawing Algorithm  
put the values of x1 and y1  
50  
200  
put the values of x2 and y2  
250  
200
```



```
Digital Differential Line Drawing Algorithm  
put the values of x1 and y1  
300  
100  
put the values of x2 and y2  
100  
300
```



LAB 2:-

//BLA

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <math.h>

void main()

{

int gd = DETECT, gm, x1, y1, x2, y2, lx, ly, dx, dy, pk, i;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

printf("put the values of x1 and y1\n");

scanf("%d%d", &x1, &y1);

printf("put the values of x2 and y2\n");

scanf("%d%d", &x2, &y2);

dx = abs(x2 - x1);

dy = abs(y2 - y1);

if (x2 > x1)

{

lx = 1;

}

else

{

lx = -1;

}

if (y2 > y1)

{

ly = 1;

}

else

{

ly = -1;

}

putpixel(x1, y1, RED);

if (dx > dy)

{

pk = 2 * dy - dx;

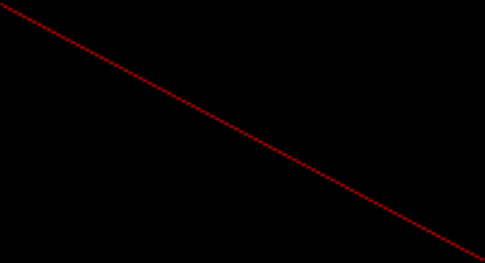
for (i = 0; i < dx; i++)

{

```
        if (pk < 0)
        {
            x1 = x1 + lx;
            y1 = y1;
            pk = pk + 2 * dy;
        }
        else
        {
            x1 = x1 + lx;
            y1 = y1 + ly;
            pk = pk + 2 * dy - 2 * dx;
        }
        putpixel(x1, y1, RED);
    }
}
else
{
    pk = 2 * dx - dy;
    for (i = 0; i < dy; i++)
    {
        if (pk < 0)
        {
            x1 = x1;
            y1 = y1 + ly;
            pk = pk + 2 * dx;
        }
        else
        {
            x1 = x1 + lx;
            y1 = y1 + ly;
            pk = pk + 2 * dx - 2 * dy;
        }
        putpixel(x1, y1, RED);
    }
}
getch();
closegraph();
}
```

OUTPUT OF LAB 2:-

```
put the values of x1 and y1
100
100
put the values of x2 and y2
300
200
```

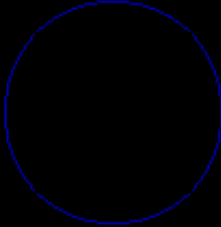


LAB 3:-Midpoint circle algorithm

```
#include<conio.h>
#include<graphics.h>
void main()
{
int x,y,x_mid,y_mid,r,d;
int g_mode,g_driver=DETECT;
clrscr();
initgraph(&g_driver,&g_mode,"C:\\TURBOC3\\BGI");
printf("***** MID POINT Circle drawing algorithm *****\n\n");
printf("\nenter the coordinates= ");
scanf("%d %d",&x_mid,&y_mid);
printf("\n now enter the radius =");
scanf("%d",&r);
x=0;
y=r;
d=1-r;
do
{
putpixel(x_mid+x,y_mid+y,1);
putpixel(x_mid+y,y_mid+x,1);
putpixel(x_mid-y,y_mid+x,1);
putpixel(x_mid-x,y_mid+y,1);
putpixel(x_mid-x,y_mid-y,1);
putpixel(x_mid-y,y_mid-x,1);
putpixel(x_mid+y,y_mid-x,1);
putpixel(x_mid+x,y_mid-y,1);
if(d<0) {
d+=(2*x)+1;
}
else{
y=y-1;
d+=(2*x)-(2*y)+1;
}
x=x+1;
}while(y>x);
getch();
}
```

OUTPUT OF LAB 3:-

```
***** MID POINT Circle drawing algorithm *****  
  
enter the coordinates= 200  
200  
  
now enter the radius =50
```



LAB 4:-

//4)Mid point ellipse algorithm

```
#include<stdio.h>
#include<graphics.h>
void main(){
    long x,y,x_center,y_center;
    long a_sqr,b_sqr, fx,fy, d,a,b,tmp1,tmp2;
    int g_driver=DETECT,g_mode;
    clrscr();

    initgraph(&g_driver,&g_mode,"C:\\TURBOC3\\BGI");
    printf("***** MID POINT ELLIPSE ALGORITHM *****");
    printf("\n\n Enter coordinate x and y = ");
    scanf("%ld%ld",&x_center,&y_center);
    printf("\n Now enter constants a and b = ");
    scanf("%ld%ld",&a,&b);
    x=0;
    y=b;
    a_sqr=a*a;
    b_sqr=b*b;
    fx=2*b_sqr*x;
    fy=2*a_sqr*y;
    d=b_sqr-(a_sqr*b)+(a_sqr*0.25);
    do
    {
        putpixel(x_center+x,y_center+y,1);
        putpixel(x_center-x,y_center-y,1);
        putpixel(x_center+x,y_center-y,1);
        putpixel(x_center-x,y_center+y,1);

        if(d<0)
        {
            d=d+fx+b_sqr;
        }
        else
        {
            y=y-1;
        }
    }
}
```



```
d=d+fx+-fy+b_sqr;
fy=fy-(2*a_sqr);
}
x=x+1;
fx=fx+(2*b_sqr);
delay(10);

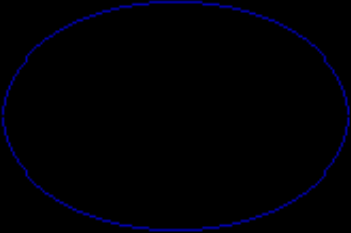
}
while(fx<fy);
tmp1=(x+0.5)*(x+0.5);
tmp2=(y-1)*(y-1);
d=b_sqr*tmp1+a_sqr*tmp2-(a_sqr*b_sqr);
do
{
putpixel(x_center+x,y_center+y,1);
putpixel(x_center-x,y_center-y,1);
putpixel(x_center+x,y_center-y,1);
putpixel(x_center-x,y_center+y,1);

if(d>=0)
d=d-fy+a_sqr;
else

{
x=x+1;
d=d+fx-fy+a_sqr;
fx=fx+(2*b_sqr);
}
y=y-1;
fy=fy-(2*a_sqr);
}
while(y>0);
getch();
closegraph();
}
```

OUTPUT OF LAB 4:-

```
***** MID POINT ELLIPSE ALGORITHM *****  
  
Enter coordinate x and y = 200  
200  
  
Now enter constants a and b = 80  
50
```



LAB 5:-

```
// Reflection
#include <conio.h>
#include <graphics.h>
#include <stdio.h>

// Driver Code
void main()
{
    // Initialize the drivers
    int gm, gd = DETECT, ax, x1 = 100;
    int x2 = 100, x3 = 200, y1 = 100;
    int y2 = 200, y3 = 100;

    // Add in your BGI folder path
    // like below initgraph(&gd, &gm,
    // "C:\\TURBOC3\\BGI");
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    cleardevice();

    // Draw the graph
    line(getmaxx() / 2, 0, getmaxx() / 2,
        getmaxy());
    line(0, getmaxy() / 2, getmaxx(),
        getmaxy() / 2);

    // Object initially at 2nd quadrant
    printf("Before Reflection Object"
        " in 2nd Quadrant");

    // Set the color
    setcolor(14);
    line(x1, y1, x2, y2);
    line(x2, y2, x3, y3);
    line(x3, y3, x1, y1);
    getch();

    // After reflection
```

```
printf("\nAfter Reflection");

// Reflection along origin i.e.,
// in 4th quadrant
setcolor(4);
line(getmaxx() - x1, getmaxy() - y1,
      getmaxx() - x2, getmaxy() - y2);

line(getmaxx() - x2, getmaxy() - y2,
      getmaxx() - x3, getmaxy() - y3);

line(getmaxx() - x3, getmaxy() - y3,
      getmaxx() - x1, getmaxy() - y1);

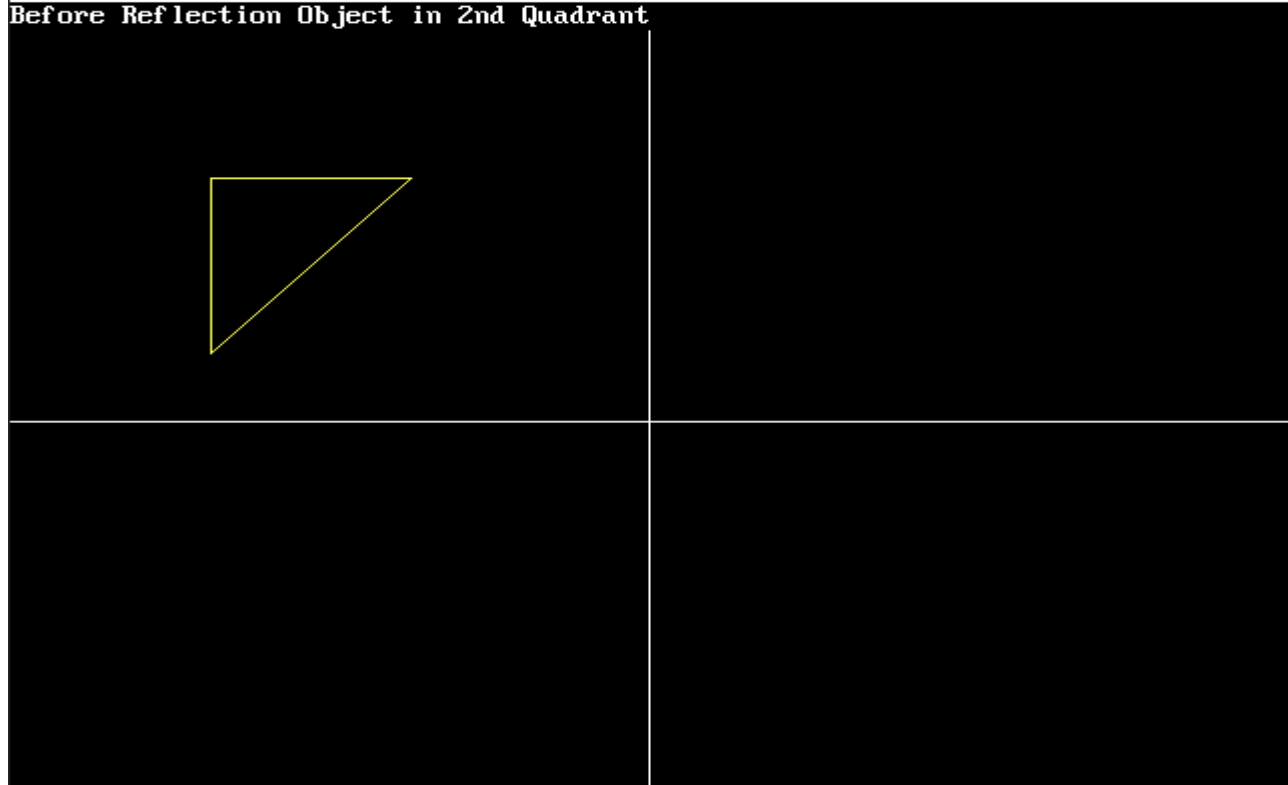
// Reflection along x-axis i.e.,
// in 1st quadrant
setcolor(3);
line(getmaxx() - x1, y1,
      getmaxx() - x2, y2);
line(getmaxx() - x2, y2,
      getmaxx() - x3, y3);
line(getmaxx() - x3, y3,
      getmaxx() - x1, y1);

// Reflection along y-axis i.e.,
// in 3rd quadrant
setcolor(2);
line(x1, getmaxy() - y1, x2,
      getmaxy() - y2);
line(x2, getmaxy() - y2, x3,
      getmaxy() - y3);
line(x3, getmaxy() - y3, x1,
      getmaxy() - y1);
getch();

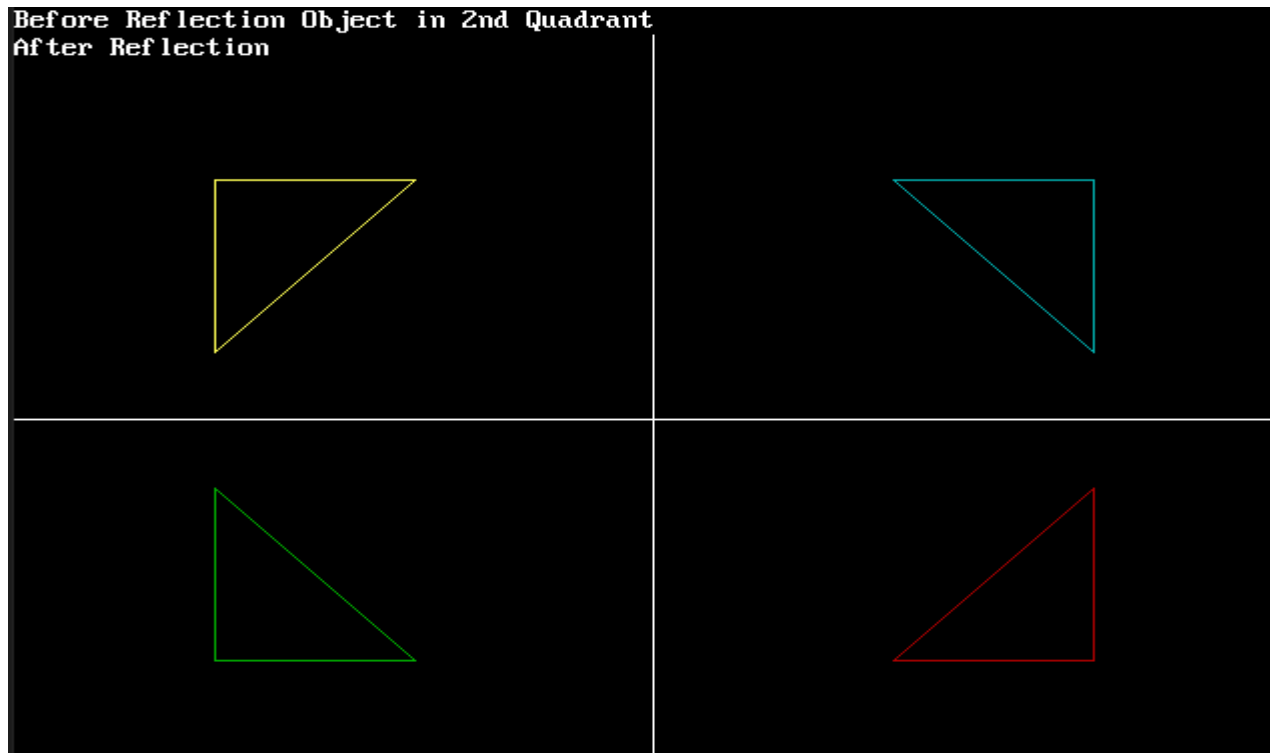
// Close the graphics
closegraph();
}
```

OUTPUT OF LAB 5:-

Before Reflection Object in 2nd Quadrant



Before Reflection Object in 2nd Quadrant
After Reflection



LAB 6:-

// C program for rotating a 2D triangle using rotation transformation techniques:

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
void DrawTriangle(int x1, int y1, int x2, int y2, int x3, int y3);
void RotateTriangle(int x1, int y1, int x2, int y2, int x3, int y3, float angle);
int main()
{
    int gd = DETECT, gm;
    int x1, y1, x2, y2, x3, y3;
    float angle;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    printf("Enter the 1st point for the triangle (x1 y1): ");
    scanf("%d%d", &x1, &y1);
    printf("Enter the 2nd point for the triangle (x2 y2): ");
    scanf("%d%d", &x2, &y2);
    printf("Enter the 3rd point for the triangle (x3 y3): ");
    scanf("%d%d", &x3, &y3);
    DrawTriangle(x1, y1, x2, y2, x3, y3);
    printf("Enter the angle for rotation (in degrees): ");
    scanf("%f", &angle);
    RotateTriangle(x1, y1, x2, y2, x3, y3, angle);
    getch();
    closegraph();
    return 0;
}
```

```
void DrawTriangle(int x1, int y1, int x2, int y2, int x3, int y3)
```

```
{
    line(x1, y1, x2, y2);

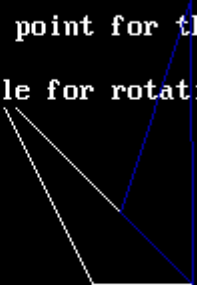
    line(x2, y2, x3, y3);

    line(x3, y3, x1, y1);
}
```

```
}  
void RotateTriangle(int x1, int y1, int x2, int y2, int x3, int y3, float angle)  
{  
    int p = x2, q = y2;  
    float radianAngle = (angle * 3.14) / 180.0;  
    int a1 = p + (x1 - p) * cos(radianAngle) - (y1 - q) * sin(radianAngle);  
    int b1 = q + (x1 - p) * sin(radianAngle) + (y1 - q) * cos(radianAngle);  
    int a2 = p + (x2 - p) * cos(radianAngle) - (y2 - q) * sin(radianAngle);  
    int b2 = q + (x2 - p) * sin(radianAngle) + (y2 - q) * cos(radianAngle);  
    int a3 = p + (x3 - p) * cos(radianAngle) - (y3 - q) * sin(radianAngle);  
    int b3 = q + (x3 - p) * sin(radianAngle) + (y3 - q) * cos(radianAngle);  
    setcolor(1);  
    DrawTriangle(a1, b1, a2, b2, a3, b3);  
}
```

OUTPUT OF LAB 6:-

```
Enter the 1st point for the triangle (x1 y1): 100
100
Enter the 2nd point for the triangle (x2 y2): 200
200
Enter the 3rd point for the triangle (x3 y3): 150
200
Enter the angle for rotation (in degrees): 45
```

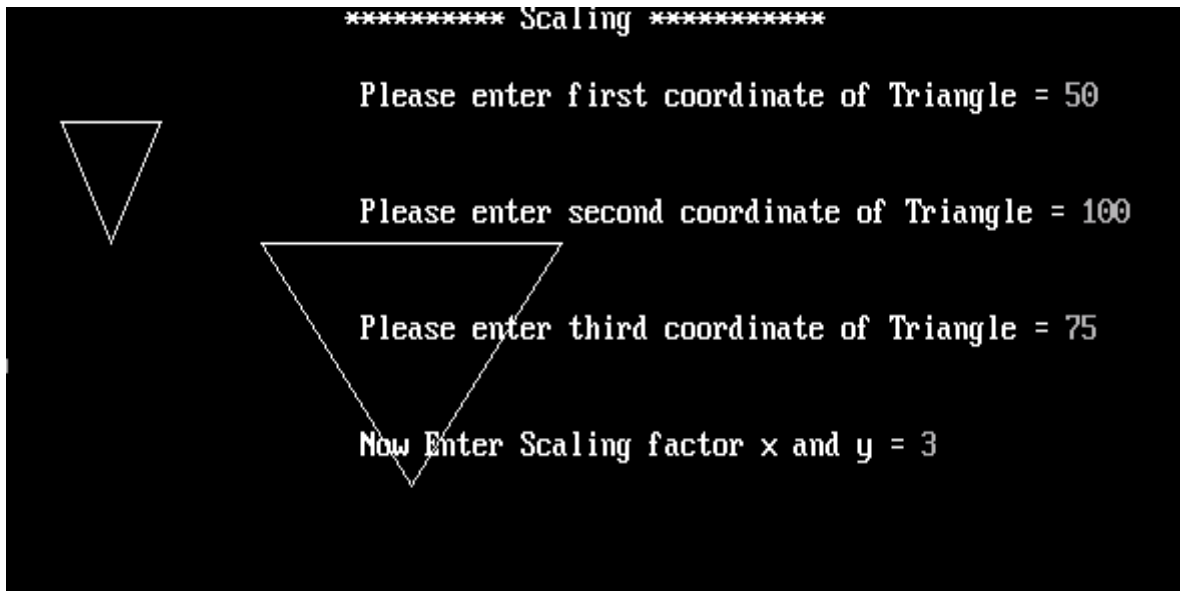


LAB 7:-

```
//Scaling of triangle
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int x,y,x1,y1,x2,y2;
int scl_fctr_x,scl_fctr_y;
int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
printf("\t\t\t***** Scaling *****\n");
printf("\n\t\t\t Please enter first coordinate of Triangle = ");
scanf("%d %d",&x,&y);
printf("\n\t\t\t Please enter second coordinate of Triangle = ");
scanf("%d %d",&x1,&y1);
printf("\n\t\t\t Please enter third coordinate of Triangle = ");
scanf("%d %d",&x2,&y2);
line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);
printf("\n\t\t\t Now Enter Scaling factor x and y = ");
scanf("%d %d",&scl_fctr_x,&scl_fctr_y);
x = x* scl_fctr_x;
x1 = x1* scl_fctr_x;
x2 = x2* scl_fctr_x;
y = y* scl_fctr_y;
y1 = y1* scl_fctr_y;
y2= y2 * scl_fctr_y ;

line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);
getch();
closegraph();
}
```

OUTPUT OF LAB 7:-



LAB 8:-

//Shearing of triangle

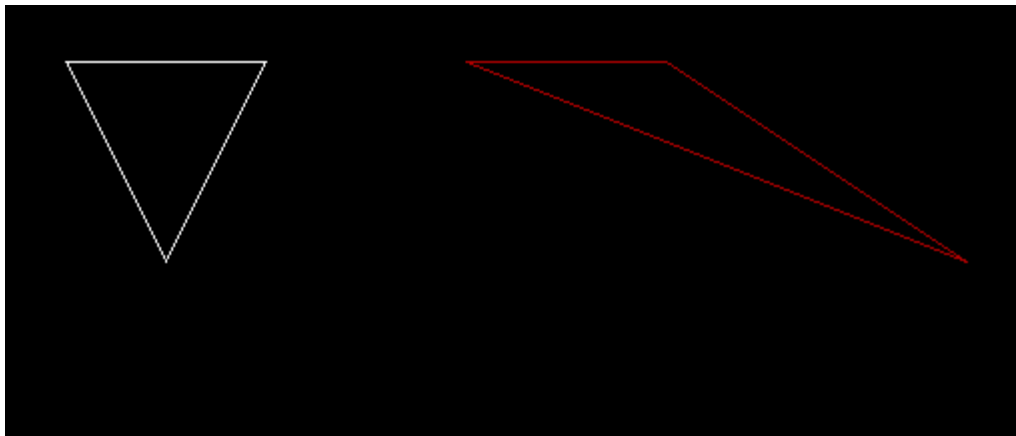
```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
void main()
{
int gd=DETECT,gm;
int x,y,x1,y1,x2,y2,shear_f;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
printf("\n please enter first coordinate = ");
scanf("%d %d",&x,&y);
printf("\n please enter second coordinate = ");
scanf("%d %d",&x1,&y1);
printf("\n please enter third coordinate = ");
scanf("%d %d",&x2,&y2);
printf("\n please enter shearing factor x = ");
scanf("%d",&shear_f);
cleardevice();
line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);

setcolor(RED);
x=x+ y*shear_f;
x1=x1+ y1*shear_f;
x2=x2+ y2*shear_f;

line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);
getch();
closegraph();
}
```

OUTPUT OF LAB 8:-

```
please enter first coordinate = 100  
100  
please enter second coordinate = 200  
100  
please enter third coordinate = 150  
200  
please enter shearing factor  $\times$  = 2
```



LAB 9:-

//Translation of a triangle.

```
#include<conio.h>
```

```
#include<graphics.h>
```

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int gd=DETECT,gm;
```

```
int x,y,x1,y1,x2,y2,tx,ty;
```

```
clrscr();
```

```
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
```

```
printf("\n Please enter first coordinate of the triangle= ");
```

```
scanf("%d %d", &x,&y);
```

```
printf("\n Enter second coordinate of the triangle = ");
```

```
scanf("%d %d",&x1,&y1);
```

```
printf("\n Enter third coordinate of the triangle = ");
```

```
scanf("%d %d",&x2,&y2);
```

```
printf("\n\t\t***** TRIANGLE before & after translation *****");
```

```
line(x,y,x1,y1);
```

```
line(x1,y1,x2,y2);
```

```
line(x2,y2,x,y);
```

```
printf("\n Now enter the translation vector = ");
```

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

```
setcolor(RED);
```

```
line(x+tx,y+ty,x1+tx,y1+ty);
```

```
line(x1+tx,y1+ty,x2+tx,y2+ty);
```

```
line(x2+tx,y2+ty,x+tx,y+ty);
```

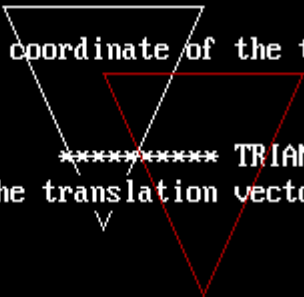
```
getch();
```

```
closegraph();
```

```
}
```

OUTPUT OF LAB 9:-

```
Please enter first coordinate of the triangle= 100
100
Enter second coordinate of the trinagle = 200
100
Enter third coordinate of the triangle = 150
200
***** TRIANGLE before & after translation *****
Now enter the translation vector = 50
30
```



LAB 10:-

```
// clipping
#include <graphics.h>
#include <stdio.h>

void cohenSutherland(int x1, int y1, int x2, int y2, int xmin, int ymin, int xmax, int ymax);

int main()
{
    int gd = DETECT, gm;
    int xmin, ymin, xmax, ymax;
    int x1, y1, x2, y2;

    // Initialize graphics mode
    initgraph(&gd, &gm, "C:\\\\TURBOC3\\\\BGI");

    // Check for successful graphics initialization
    if (graphresult() != 0)
    {
        printf("Graphics initialization failed!\n");
        return 1;
    }

    printf("Enter window coordinates (xmin ymin xmax ymax): ");
    scanf("%d%d%d%d", &xmin, &ymin, &xmax, &ymax);

    // Draw the clipping window
    rectangle(xmin, ymin, xmax, ymax);

    printf("Enter line coordinates (x1 y1 x2 y2): ");
    scanf("%d%d%d%d", &x1, &y1, &x2, &y2);

    // Draw the original line
    line(x1, y1, x2, y2);

    delay(1000);
    cleardevice();
}
```

```
// Perform Cohen-Sutherland line clipping  
cohenSutherland(x1, y1, x2, y2, xmin, ymin, xmax, ymax);
```

```
getch();  
closegraph();
```

```
return 0;  
}
```

```
void cohenSutherland(int x1, int y1, int x2, int y2, int xmin, int ymin, int xmax, int  
ymax)
```

```
{  
    int code1 = 0, code2 = 0, codeOut;  
    int accept = 0, done = 0;  
    int x, y;  
  
    while (!done)  
    {  
        code1 = 0;  
        code2 = 0;  
  
        if (x1 < xmin)  
            code1 |= 1; // to the left of clip window  
        if (x1 > xmax)  
            code1 |= 2; // to the right of clip window  
        if (y1 < ymin)  
            code1 |= 4; // below the clip window  
        if (y1 > ymax)  
            code1 |= 8; // above the clip window  
  
        if (x2 < xmin)  
            code2 |= 1; // to the left of clip window  
        if (x2 > xmax)  
            code2 |= 2; // to the right of clip window  
        if (y2 < ymin)  
            code2 |= 4; // below the clip window  
        if (y2 > ymax)
```


code2 |= 8; // above the clip window

codeOut = code1 | code2;

```
if (codeOut == 0)
{
    accept = 1;
    done = 1;
}
else if (code1 & code2)
{
    done = 1;
}
else
{
    if (code1 != 0)
    {
        codeOut = code1;
    }
    else
    {
        codeOut = code2;
    }

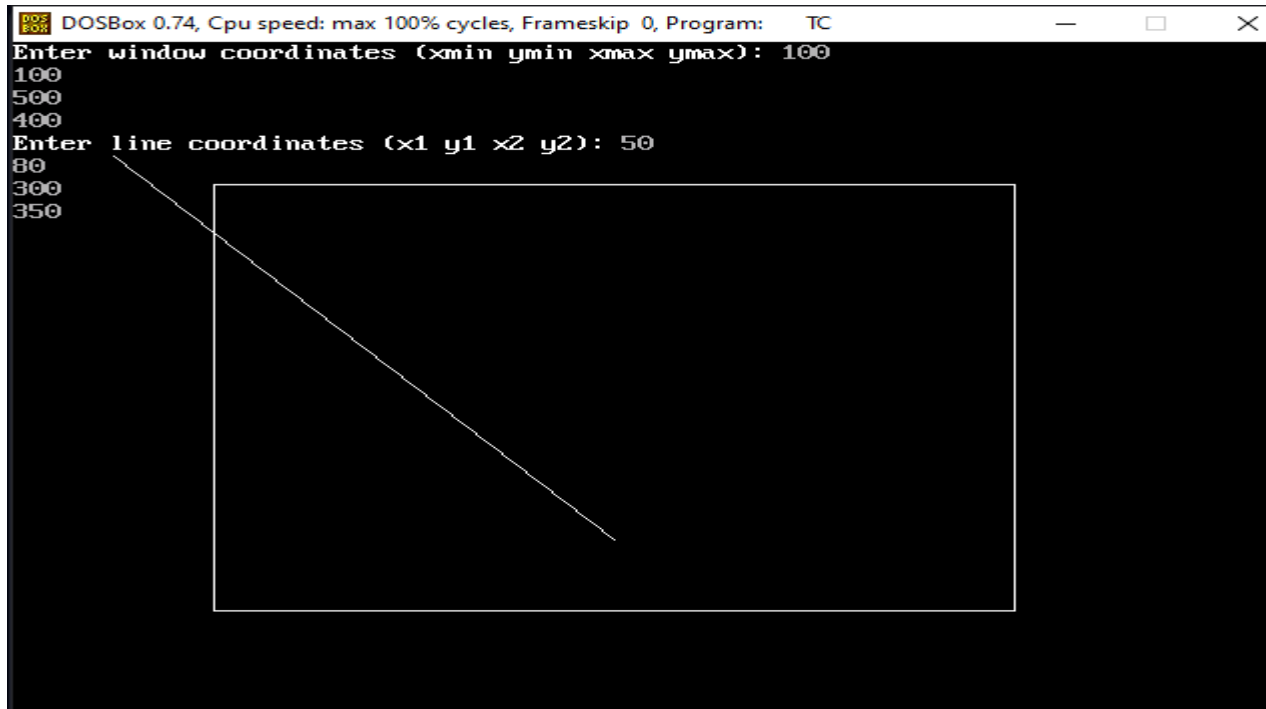
    if (codeOut & 1)
    {
        y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1);
        x = xmin;
    }
    else if (codeOut & 2)
    {
        y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1);
        x = xmax;
    }
    else if (codeOut & 4)
    {
        x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1);
        y = ymin;
    }
}
```

```
}
else
{
    x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1);
    y = ymax;
}

if (codeOut == code1)
{
    x1 = x;
    y1 = y;
}
else
{
    x2 = x;
    y2 = y;
}
}

if (accept)
{
    setcolor(RED);
    line(x1, y1, x2, y2);
    delay(1000);
}
}
```

LAB 10 OUTPUT:-



CLIPPED LINE:-

