

KCT COLLEGE OF ENGG AND TECHNOLOGY

DEPTT. OF COMPUTER SCIENCE AND ENGG.

LAB MANUAL

Subject: Computer Graphics(BTCS 509)

Semester- 5th



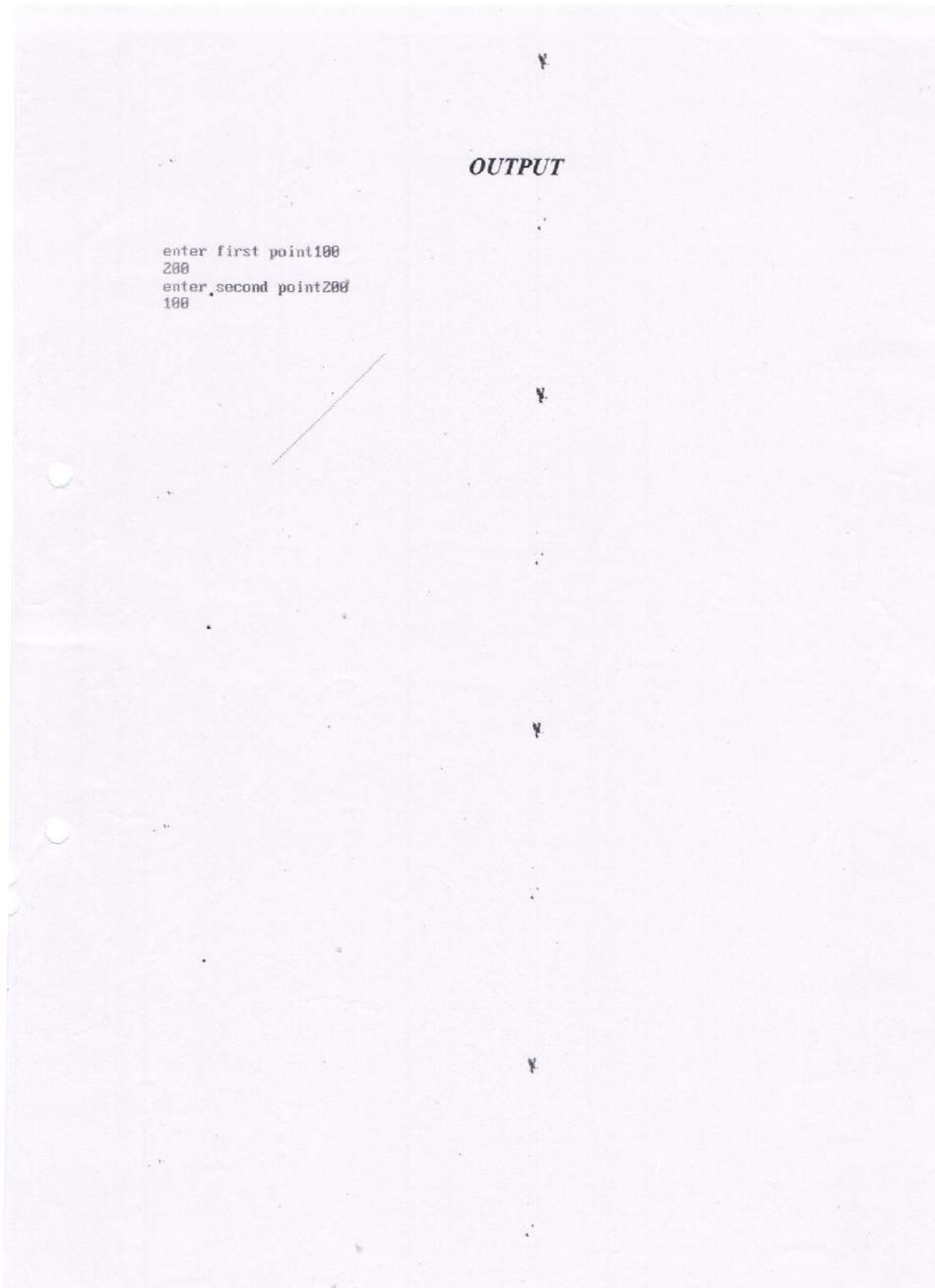
INDEX

Sr.No.	Name of Experiment
1.	WAP to draw a straight line using CDA algorithm.
2.	WAP to draw a straight line using Bresenham's algorithm
3.	WAP to draw an cilipse using mid-point ellipse drawing algorithm.
4.	WAP to draw a circle using Bresenham's algorithm.
5.	WAP to show line clipping.
6.	WAP to rotate a triangle about origin.
7.	Program to scale the triangle.
8.	Program to translate a triangle.
9.	Program to rotate a point about a point.
10.	Program to rotate a point about origin.
11.	Program to reflect a triangle.
12.	Program to fill a polygon.

1. WAP TO DRAW A LINE USING DDA ALGORITHM.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
    int x,y,x1,x2,y1,y2,k,dx,dy,s,xi,yi;
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"C:\\tc\\bgi.");
    printf("enter first point");
    scanf("%d%d",&x1,&y1);
    printf("enter second point");
    scanf("%d%d",&x2,&y2);
    x=x1;
    y=y1;
    putpixel(x,y,7);
    dx=x2-x1;
    dy=y2-y1;
    if(abs(dx)>abs(dy))
        s=abs(dx);
    else
        s=abs(dy);
    xi=dx/s;
    yi=dy/s;
    x=x1;
    y=y1;
    putpixel(x,y,7);
    for(k=0;k<s;k++)
    {
```

```
x=x+xi;  
y=y+yi;  
putpixel(x,y,7);  
}  
getch();  
closegraph();  
}
```



2. WAP TO DRAW A LINE USING MID POINT ALGORITHM OR BRESENHAM'S ALGORITHM.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
    int x,y,x1,y1,x2,y2,p,dx,dy;
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"C:\\tc\\BGI:");
    printf("\nEnter the x-coordinate of the first point ::");
    scanf("%d",&x1);
    printf("\nEnter the y-coordinate of the first point ::");
    scanf("%d",&y1);
    printf("\nEnter the x-coordinate of the second point ::");
    scanf("%d",&x2);
    printf("\nEnter the y-coordinate of the second point ::");
    scanf("%d",&y2);
    x=x1;
    y=y1;
    dx=x2-x1;
    dy=y2-y1;
    putpixel(x,y,2);
    p=(2dy-dx);
    while(x<=x2)
    {
        if(p<0)
        {
            x=x+1;
```

```
        p=2*x-dx;
    }
    else
    {
        x=x+1;
        y=y+1;
        p=p+2*dy;
    }
    putpixel(x,y,7);
}
getch();
closegraph();
}
```

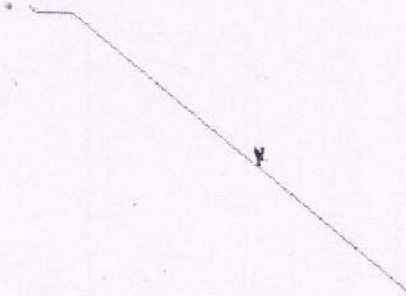
OUTPUT

Enter the x-coordinate of the first point ::100

Enter the y-coordinate of the first point ::250

Enter the x-coordinate of the second point ::500

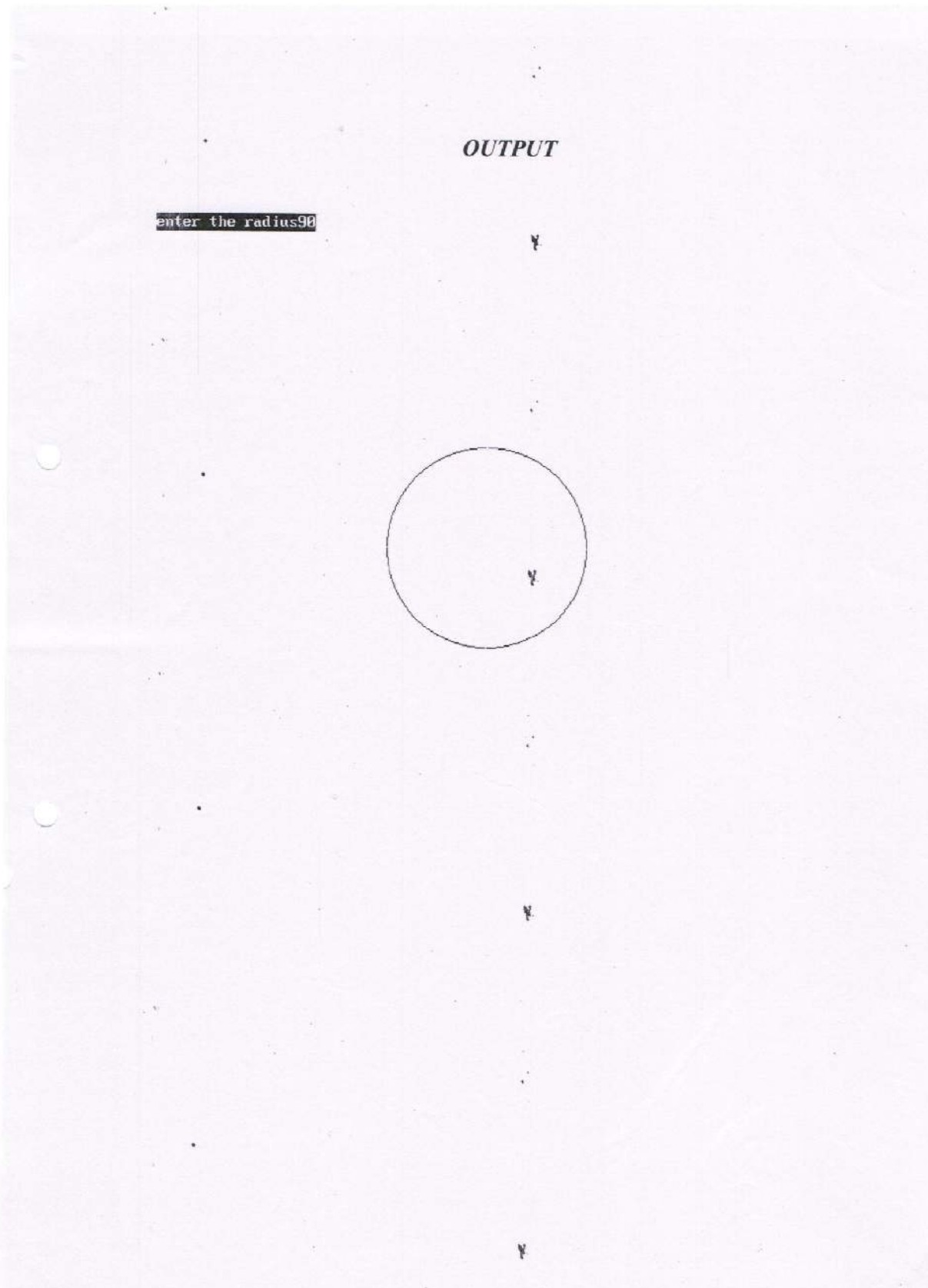
Enter the y-coordinate of the second point ::600



2) WAP TO DRAW A CIRCLE USING BRESENHAM'S ALGORITHM.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void circlepoints(int,int);
void main()
{
    int x,y,p,r;
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"C:\\tc\\bgi.");
    clrscr();
    printf("enter the radius");
    scanf("%d",&r);
    x=0;y=r;p=1-r;
    while(x<y)
    {
        x++;
        if(p>0)
        {
            p=p+2*(x-y)+1;
            y--;
        }
        else
            p=p+2*x+1;
        circlepoints(x,y);
    }
    getch();
    closegraph();
}
void circlepoints(int x,int y)
```

```
putpixel(x+300,y+300,8);  
putpixel(x+300,-y+300,8);  
putpixel(-x+300,y+300,8);  
putpixel(-x+300,-y+300,8);  
putpixel(y+300,x+300,8);  
putpixel(y+300,-x+300,8);  
putpixel(-y+300,x+300,8);  
putpixel(-y+300,-x+300,8);
```



**3. WAP TO DRAW AN ELLIPSE USING MID-POINT ELLIPSE
DRAWING ALGORITHM.**

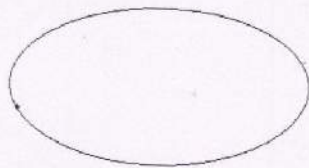
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void ellips(int x,int y);
void completellipse(int r,int g,int u,int v)
{
    float s,k,e,f,x;
    double p1,p2;
    s=r;k=g;
    e=(pow((s+.5),2));
    f=(pow((k-1),2));
    p2=((u*e)+(v*f)-(u*v));
    ellips(s,k);
    while(k>=0)
    {
        if(p2>0)
            p2=(p2+v-(2*v*s));
        else
        {
            p2=(p2+(2*u*(s+1))-(2*v*(k-1))+v);
            s++;
        }
        k--;
        ellips(s,k);
    }
}
```

```
void main()
{
    int gdriver=DETECT,gmode;
    int a,b,x,y;
    long u,v,p1;
    initgraph(&gdriver,&gmode,"C:\\tc\\bgi:");
    printf("\n enter the length of major axis:");
    scanf("%d",&a);
    printf("\n enter the length of minor axis:");
    scanf("%d",&b);
    x=0;
    y=b;
    u=pow(b,2);
    v=pow(a,2);
    p1=(u-(v*b)+(.25*v));
    ellips(x,y);
    while(2*(u*x)<=2*(v*y))
    {
        x++;
        if(p1<0)
            p1=(p1+(2*u*v)+v);
        else
        {
            p1=(p1+(2*u*x)-(2*v*y)+u);
            y--;
        }
        ellips(x,y);
    }
    completellipse(x,y,u,v);
    getch();
    closegraph();
}
```

```
}  
void ellips(int x,int y)  
{  
    putpixel(x+200,y+200,8);  
    putpixel(-x+200,y+200,8);  
    putpixel(x+200,-y+200,8);  
    putpixel(-x+200,-y+200,8);  
}
```

OUTPUT

enter the length of major axis:100
enter the length of minor axis:50



5. WAP TO SHOW LINE CLIPPING.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void storepoints(int,int,int,int,int,int,int[]);
void main()
{
    int gdriver=DETECT,gmode;
    int x1,x2,y1,y2,xmax,ymax,xmin,ymin,a[10],b[10],xi1,xi2,yi1,yi2,flag=0;
    float m;
    int i;
    clrscr();

    printf("output");
    printf("\n");
    printf("enter the value of x1,y1,x2,y2: __>");
    scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
    printf("enter the value of xmax,ymax,xmin,ymin:");
    scanf("%d%d%d%d",&xmax,&ymax,&xmin,&ymin);
    storepoints(x2,y2,ymin,ymax,xmax,xmin,b);
    for(i=1;i<=4;i++)
    {
        if(a[i]*b[i]==0)
            flag=1;
        else
            flag=0;
    }
    if(flag==1)
```



```
{
    m=(y2-y1)/(x2-x1);
    x1=x1;
    y1=y1;
}
if(a[1]==1)
{
    y1=ymax;
    x1=x1+((1/m)*(y1-y1));
}
else
{
    if(a[2]==1)
    {
        y1=ymin;
        x1=x1+((1/m)*(y1-y1));
    }
}
if(a[3]==1)
{
    x1=xmax;
    y1=y1+(m*(x1-x1));
}
if(a[4]==1)
{
    x1=xmin;
    y1=y1+(m*(x1-x1));
}
else
    if(b[1]==1)
    {
```

```
        yi2=ymax;
        xi2=x2+((1/m)*(yi2-y2));
    }
    else
        if(b[2]==1)
        {
            yi2=ymin;
            xi2=x2+((1/m)*(yi2-y2));
        }
    else
        if(b[3]==1)
        {
            xi2=xmax;
            yi2=y2+((1/m)*(xi2-x2));
        }
    else
        if(b[4]==1)
        {
            xi2=xmin;
            yi2=y2+(m*(xi2-x2));
        }
}

clrscr();
initgraph(&gdriver,&gmode,"c://tc//bgi:");
rectangle(xmin,ymin,xmax,ymax);
line(x1,y1,x2,y2);
delay(5000);
closegraph();
clrscr();
initgraph(&gdriver,&gmode,"c://tc//bgi:");
line(xi1,yi1,xi2,yi2);
rectangle(xmin,ymin,xmax,ymax);
```

```
    if(flag==0)
    {
        printf("\n no clipping is required");
    }
    getch();
    closegraph();
}

void storepoints(int x1,int y1,int ymax,int xmax,int xmin,int ymin,int c[10])
{
    if((y1-ymax)>0)
        c[1]=1;
    else
        c[1]=0;
    if((ymin-y1)>0)
        c[2]=1;
    else
        c[2]=0;
    if((x1-xmax)>0)
        c[3]=1;
    else
        c[3]=0;
    if((xmin-x1)>0)
        c[4]=1;
    else
        c[4]=0;
}
```

OUTPUT

enter the value of x1,y1,x2,y2: __>10

10

100

100

enter the value of xmax,ymax,xmin,ymin50

50

0

0



6. WAP TO ROTATE A TRIANGLE ABOUT ORIGIN.

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<math.h>
void main()
{
clrscr();
int graphdriver=DETECT,graphmode;
initgraph(&graphdriver,&graphmode,"...\\bgi");

int x,y,x1,a[3][3];
double b[3][3],c[3][3];
cout<<"\n    Enter 1st coordinates of triangle:";
cin>>a[0][0]>>a[1][0];

cout<<"\n    Enter 2nd coordinates of triangle:";
cin>>a[0][1]>>a[1][1];

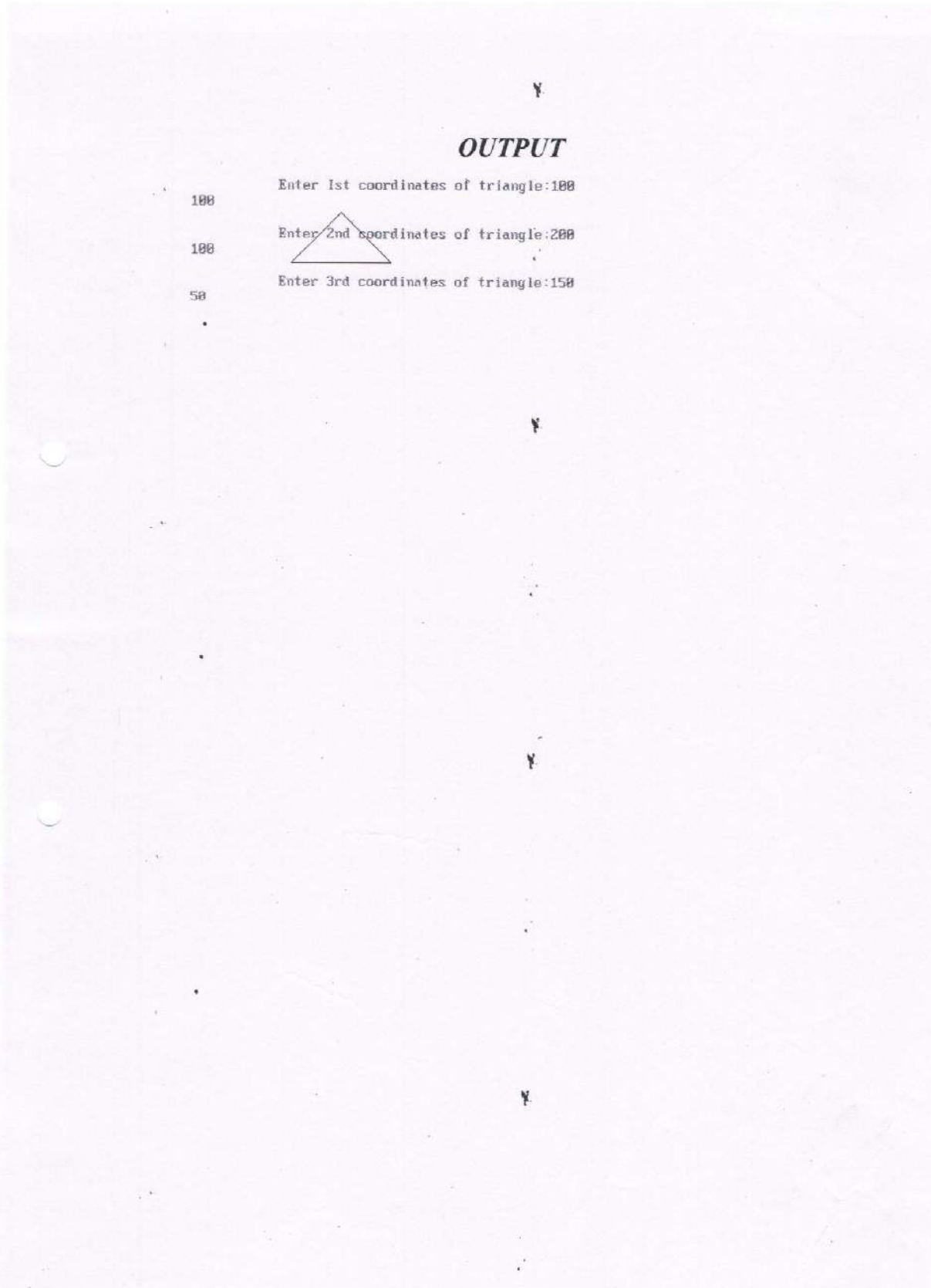
cout<<"\n    Enter 3rd coordinates of triangle:";
cin>>a[0][2]>>a[1][2];

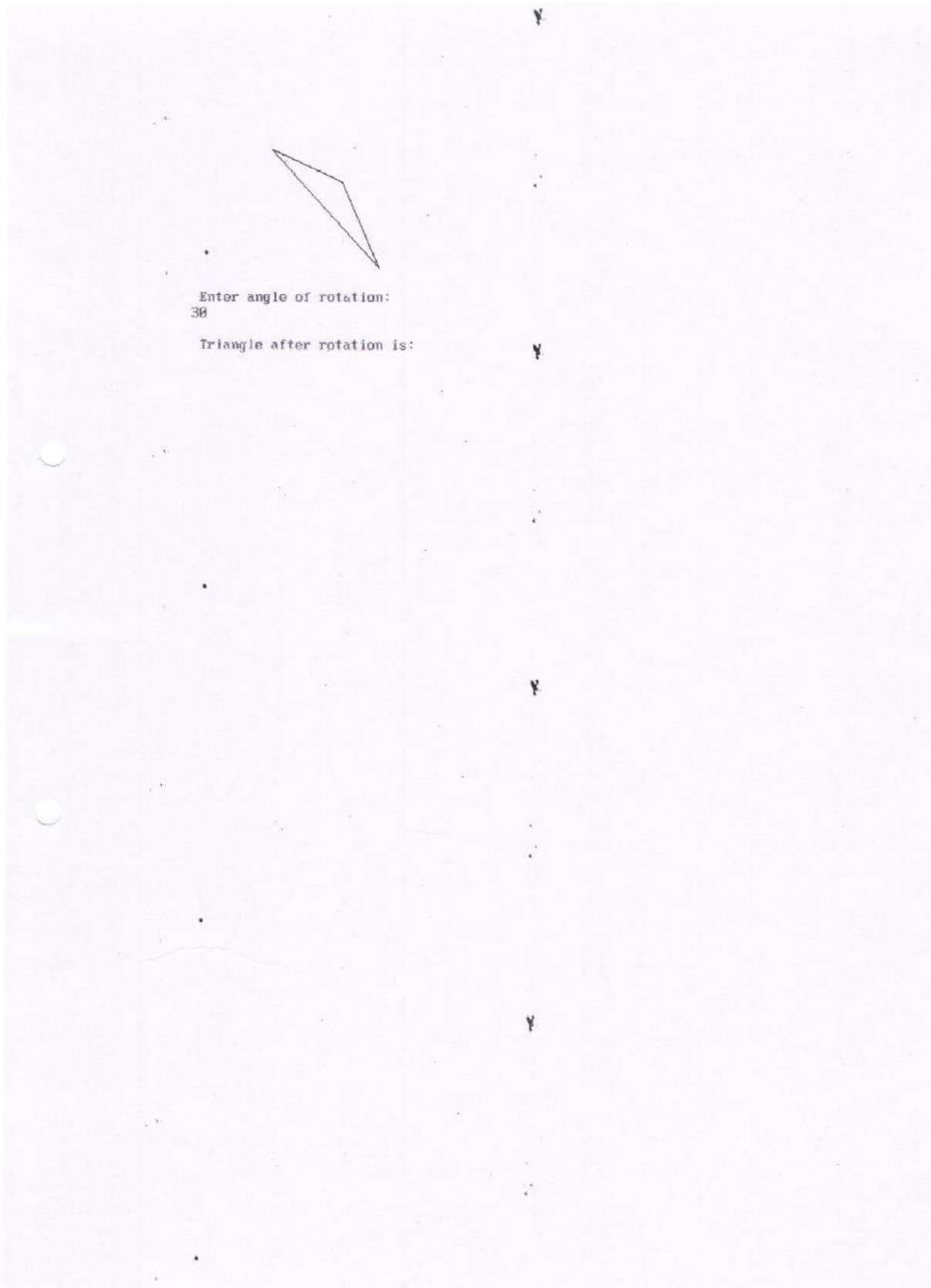
line(a[0][0],a[1][0],a[0][1],a[1][1]);
line(a[0][1],a[1][1],a[0][2],a[1][2]);
line(a[0][0],a[1][0],a[0][2],a[1][2]);
getch();
cleardevice();
cout<<"\n Enter angle of rotation:\n";
cin>>x;
```

```
b[0][0]=b[1][1]=cos((x*3.14)/180);
b[0][1]=-sin((x*3.14)/180);
b[1][0]=sin((x*3.14)/180);
b[2][2]=1;
b[2][0]=b[2][1]=b[0][2]=b[1][2]= 0;
for(int i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
c[i][j]=0;
for (int k=0; k<3;k++)
{
c[i][j]+=a[i][k]*b[k][j];
}
x1=(c[i][j]+0.5);
a[i][j]=x1;
}
}
cout<<"\n Triangle after rotation is:\n" ;

line(a[0][0],a[1][0],a[0][1],a[1][1]);
line(a[0][1],a[1][1],a[0][2],a[1][2]);
line(a[0][0],a[1][0],a[0][2],a[1][2]);

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





7. PROGRAM TO SCALE THE TRIANGLE

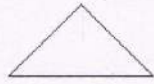
```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd, &gm, "");
cleardevice();
int x1,y1,x2,y2,x3,y3,x4,y4;
float sx,sy;
cout<<"Enter the first coordinates of triangle\n";
cin>>x1>>y1;
cout<<"Enter the second coordinates of triangle\n";
cin>>x2>>y2;
cout<<"Enter the third coordinates of triangle\n";
cin>>x3>>y3;
int poly[8]={x1,y1,x2,y2,x3,y3,x1,y1};
cleardevice();
drawpoly(4,poly);
getch();
cout<<"Enter the scaling factors\n";
cin>>sx>>sy;
x4=sx*x1-x1;
y4=sy*y1-y1;

x1=sx*x1-x4;
y1=sy*y1-y4;
x2=sx*x2-x4;
```

```
y2=sy*y2-y4;  
x3=sx*x3-x4;  
y3=sy*y3-y4;  
poly[0]=x1;  
poly[1]=y1;  
poly[2]=x2;  
poly[3]=y2;  
poly[4]=x3;  
poly[5]=y3;  
poly[6]=x1;  
poly[7]=y1;  
getch();  
cleardevice();  
drawpoly(4,poly);  
getch();  
closegraph();  
}
```

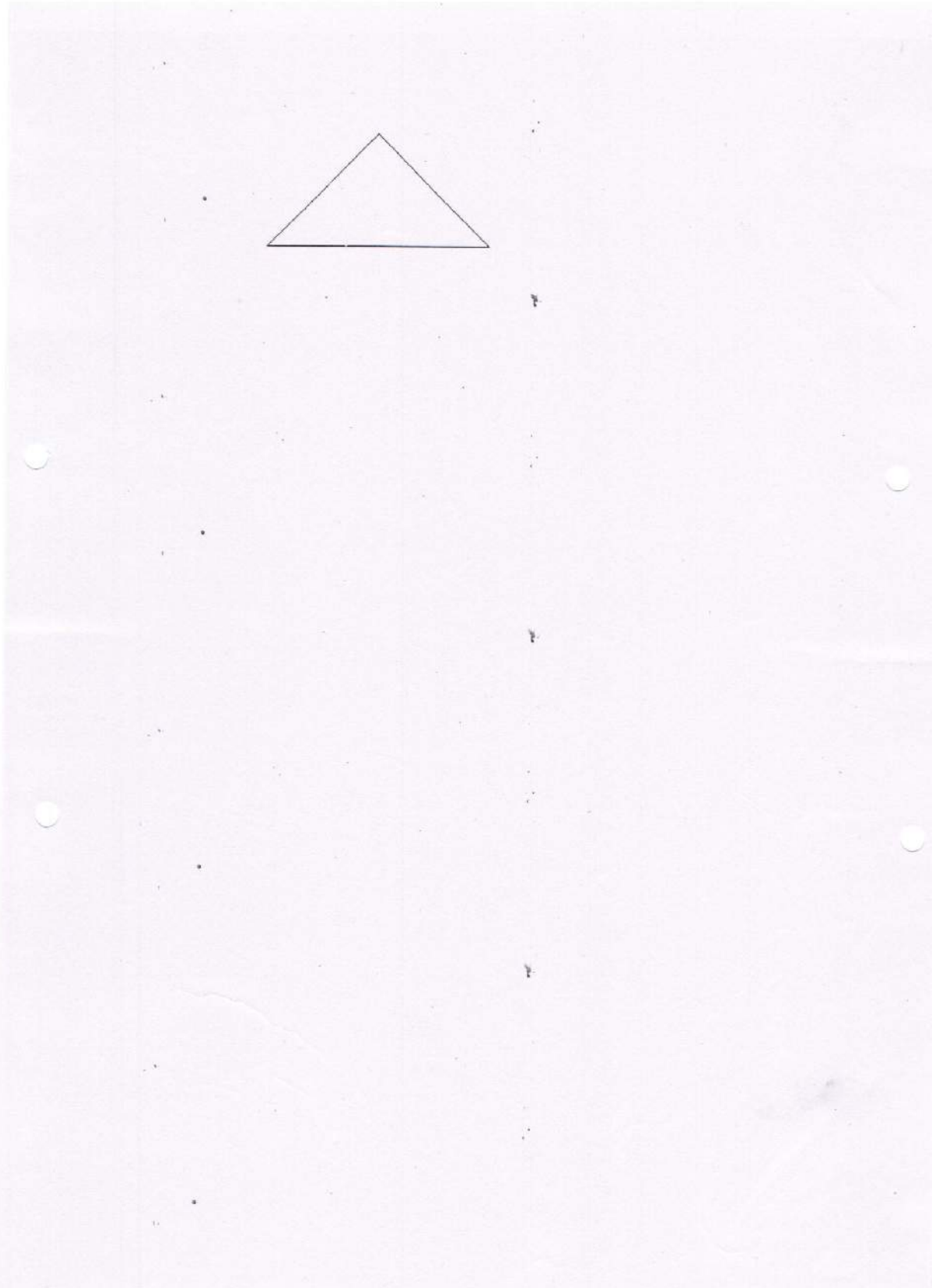
OUTPUT

```
Enter the first coordinates of triangle  
100  
100  
Enter the second coordinates of triangle  
200  
100  
Enter the third coordinates of triangle  
150  
50
```



Enter the scaling factors

2
2



8. PROGRAM TO TRANSLATE A TRIANGLE

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<math.h>

void main()
{
clrscr();
int graphdriver=DETECT,graphmode;
initgraph(&graphdriver,&graphmode,"...\\bgi");

int x,y,x1,y1,x2,y2,x3,y3;
cout<<"n    Enter 1st coordinates of triangle:";
cin>>x1>>y1;

cout<<"n    Enter 2nd coordinates of triangle:";
cin>>x2>>y2;

cout<<"n    Enter 3rd coordinates of triangle:";
cin>>x3>>y3;

cleardevice();
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x1,y1,x3,y3);
getch();
cleardevice();
```

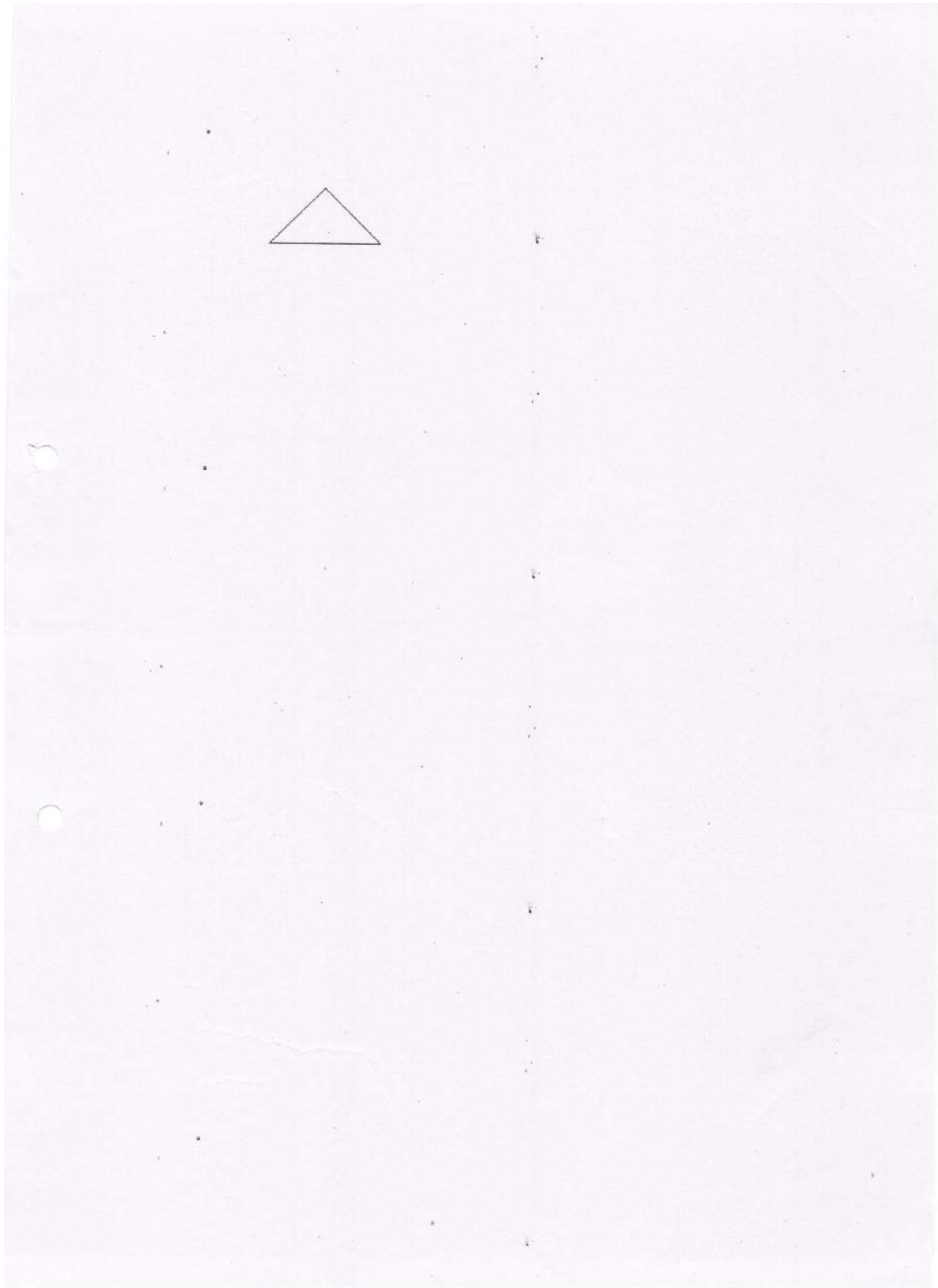
```
cout<<"\n Enter translatio factors :\n";
cin>>x>>y;

x1-=x;
y1-=y;
x2-=x;
y2-=y;
x3-=x;
y3-=y;

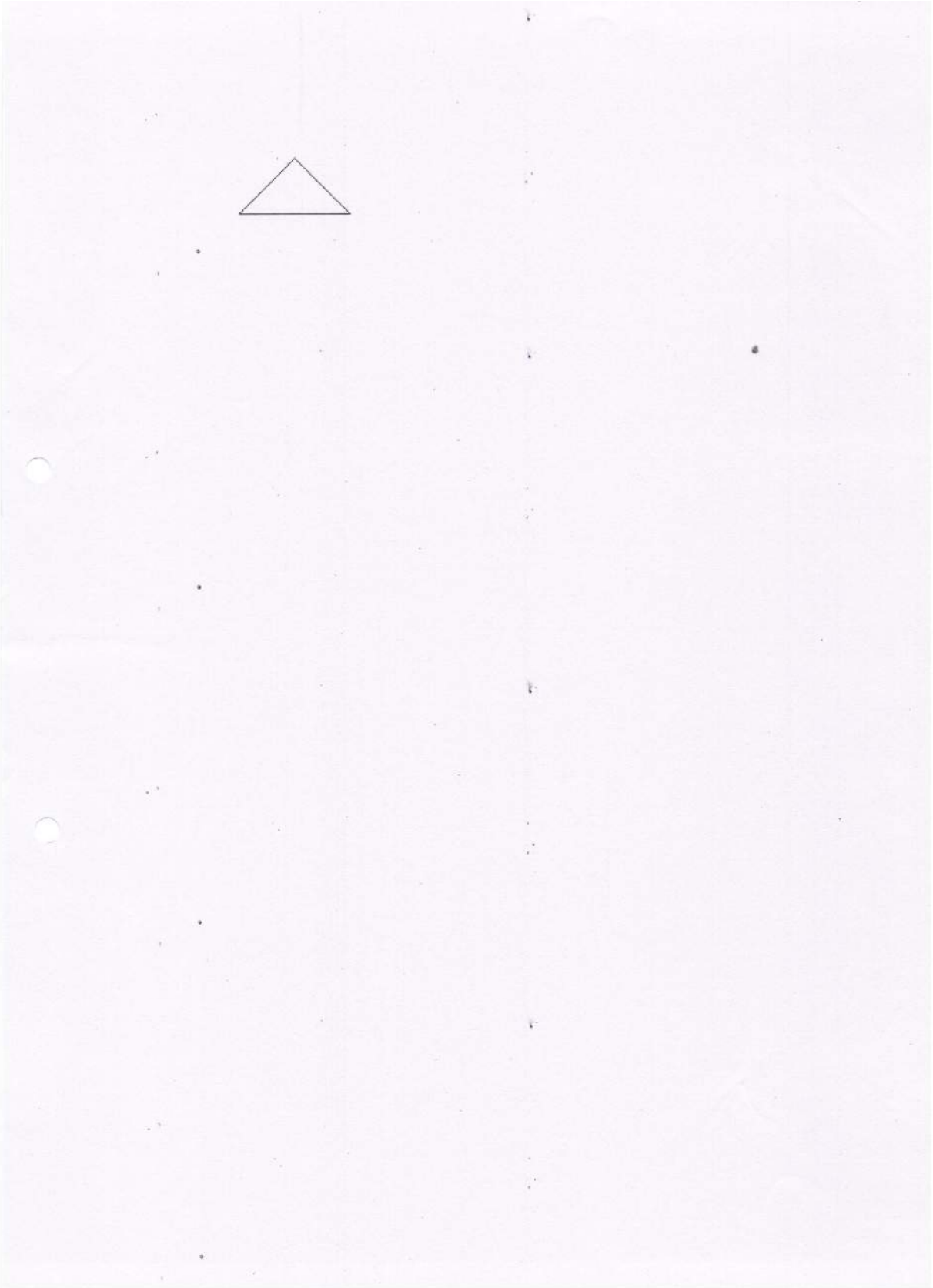
cleardevice();
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x1,y1,x3,y3);
getch();
closegraph();
}
```

OUTPUT

```
Enter 1st coordinates of triangle:100
100
Enter 2nd coordinates of triangle:200
100
Enter 3rd coordinates of triangle:150
50
```

Enter translatio factors :
20
30



9. PROGRAM TO ROTATE A POINT ABOUT A POINT

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
void main()
{
clrscr();
int gm,gd=DETECT;
initgraph(&gd,&gm,"");
int h,k,x1,y1,x2,y2,x3,y3;
float t;
cout<<" OUTPUT"<<endl;
cout<<"Enter the coordinates of point"<<endl;
cin>>x2>>y2;
putpixel(x2,y2,2);

cout<<"Enter the coordinates of point around which rotation is done"<<endl;
cin>>h>>k;
putpixel(h,k,2);

cout<<"Enter the angle for rotation"<<endl;
cin>>t;
cleardevice();
x1=(h*cos(t))-(k*sin(t));
y1=(h*sin(t))+(k*cos(t));
x3=x1+x2-h;
y3=y1+y2-k;

cout<<"Point after rotation is:";
```

```
putpixel(x3,y3,2);
```

```
getch();
```

```
closegraph();
```

```
}
```

OUTPUT

```
OUTPUT
```

```
Enter the coordinates of point
```

```
100
```

```
100
```

```
Enter the coordinates of point around which rotation is done
```

```
50
```

```
50
```

```
Enter the angle for rotation
```

```
30
```

Point after rotation is:

6. PROGRAM TO ROTATE A POINT ABOUT ORIGIN

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
void main()
{
clrscr();
int gm,gd=DETECT;
initgraph(&gd,&gm,"");
int h,k,x1,y1,x2,y2,x3,y3;
float t;
cout<<" OUTPUT"<<endl;
cout<<"Enter the coordinates of point"<<endl;
cin>>x2>>y2;
putpixel(x2,y2,2);

cout<<"Enter the angle for rotation"<<endl;
cin>>t;
cleardevice();
x1=int(x2*cos(t*3.14/180))-(y2*sin(t*3.14/180));
y1=int(x2*sin(t*3.14/180)+(y2*cos(t*3.14/180));
cout<<"Point after rotation is:";
putpixel(x1,y1,2);

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

OUTPUT

```
OUTPUT
Enter the coordinates of point
100
100
Enter the angle for rotation
30
```


Point after rotation is:

||. PROGRAM TO REFLECT A TRIANGLE

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<math.h>
void main()
{
clrscr();
int graphdriver=DETECT,graphmode;
initgraph(&graphdriver,&graphmode,"...\\bgi");

int x,y,x1,a[3][3];
double b[3][3],c[3][3];
cout<<"\n    Enter 1st coordinates of triangle:";
cin>>a[0][0]>>a[1][0];

cout<<"\n    Enter 2nd coordinates of triangle:";
cin>>a[0][1]>>a[1][1];

cout<<"\n    Enter 3rd coordinates of triangle:";
cin>>a[0][2]>>a[1][2];

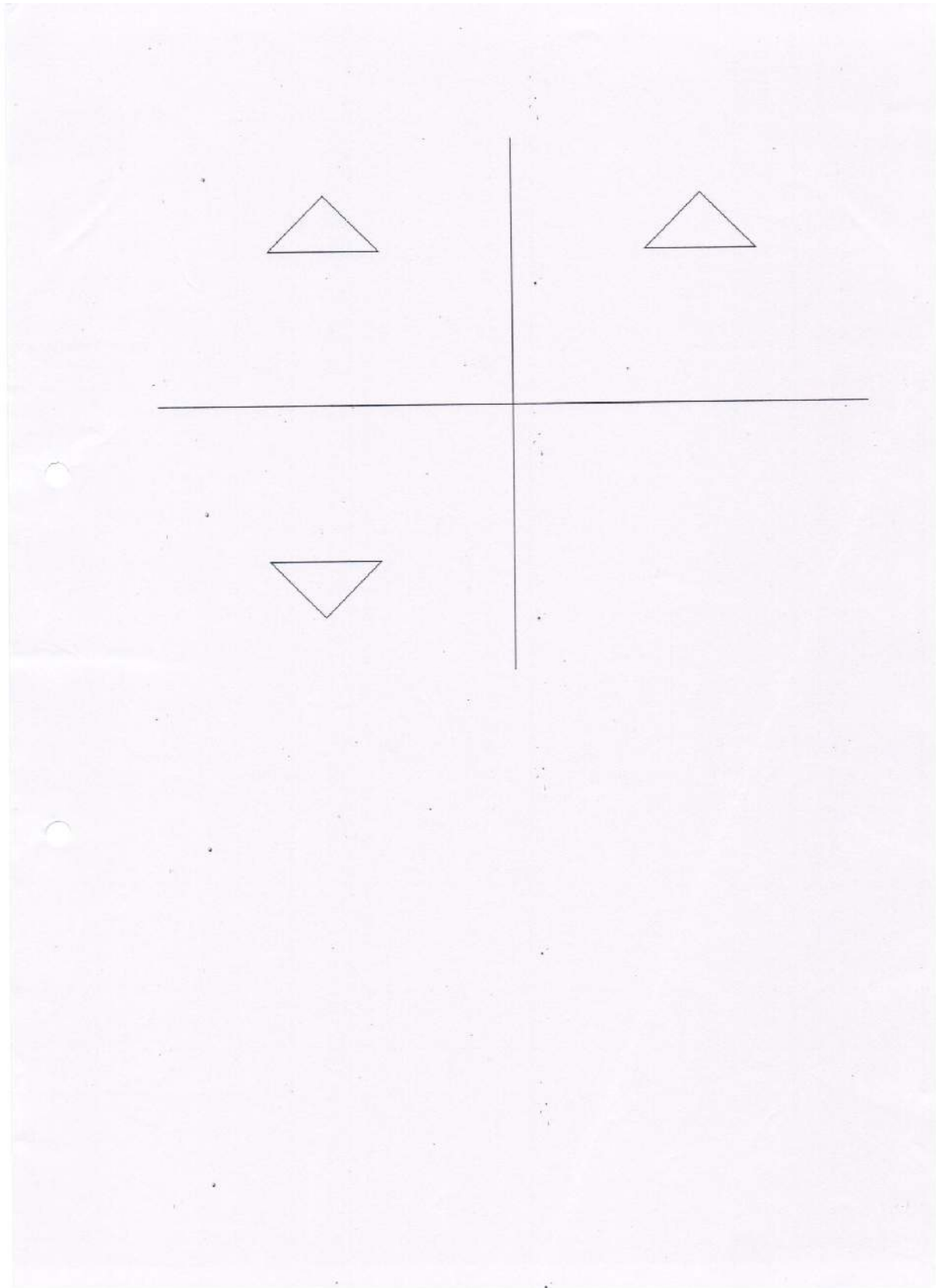
cout<<"\n Enter 1. for reflection in x-axis:\n";
cout<<"\n Enter 2. for reflection in y-axis:\n";
cout<<"\n Enter 3. for reflection in both the axis:\n";
cin>>x;
cleardevice();
line(320,0,320,479);
line(0,240,639,240);
```

```
line(a[0][0],a[1][0],a[0][1],a[1][1]);
line(a[0][1],a[1][1],a[0][2],a[1][2]);
line(a[0][0],a[1][0],a[0][2],a[1][2]);
switch(x)
{
case 1:b[0][0]=640-a[0][0];
      b[0][1]=640-a[0][1];
      b[0][2]=640-a[0][2];
      b[1][0]=a[1][0];
      b[1][1]=a[1][1];
      b[1][2]=a[1][2];
      line(320,0,320,479);
      line(0,240,639,240);
      line(b[0][0],b[1][0],b[0][1],b[1][1]);
      line(b[0][1],b[1][1],b[0][2],b[1][2]);
      line(b[0][0],b[1][0],b[0][2],b[1][2]);
      getch();
      break;
case 2:b[1][0]=480-a[1][0];
      b[1][1]=480-a[1][1];
      b[1][2]=480-a[1][2];
      b[0][0]=a[0][0];
      b[0][1]=a[0][1];
      b[0][2]=a[0][2];
      line(320,0,320,479);
      line(0,240,639,240);
      line(b[0][0],b[1][0],b[0][1],b[1][1]);
      line(b[0][1],b[1][1],b[0][2],b[1][2]);
      line(b[0][0],b[1][0],b[0][2],b[1][2]);
      getch();
      break;
```

```
case 3: b[0][0]=640-a[0][0];
        b[0][1]=640-a[0][1];
        b[0][2]=640-a[0][2];
        b[1][0]=a[1][0];
        b[1][1]=a[1][1];
        b[1][2]=a[1][2];
        line(320,0,320,479);
        line(0,240,639,240);
        line(b[0][0],b[1][0],b[0][1],b[1][1]);
        line(b[0][1],b[1][1],b[0][2],b[1][2]);
        line(b[0][0],b[1][0],b[0][2],b[1][2]);
        b[1][0]=480-a[1][0];
        b[1][1]=480-a[1][1];
        b[1][2]=480-a[1][2];
        b[0][0]=a[0][0];
        b[0][1]=a[0][1];
        b[0][2]=a[0][2];
        line(320,0,320,479);
        line(0,240,639,240);
        line(b[0][0],b[1][0],b[0][1],b[1][1]);
        line(b[0][1],b[1][1],b[0][2],b[1][2]);
        line(b[0][0],b[1][0],b[0][2],b[1][2]);
        getch();
        break;
    }
    getch();
    closegraph();
}
```

OUTPUT

```
Enter 1st coordinates of triangle:100
100
Enter 2nd coordinates of triangle:200
100
Enter 3rd coordinates of triangle:150
50
Enter 1. for reflection in x-axis:
Enter 2. for reflection in y-axis:
Enter 3. for reflection in both the axis:
3
```



19. PROGRAM TO FILL A POLYGON

```
#include<conio.h>
#include<iostream.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
#include<process.h>
void main()
{
int graphdriver=DETECT,graphmode;
initgraph(&graphdriver,&graphmode,"...\\bgi");
int p=1,x;

int a[12]={100,100,150,150,200,100,200,200,100,200,100,100};
drawpoly(6,a);

for(int i=100;i<200;i++)
{
p=1;
for(int j=100;j<=200;j++)
{
x=getpixel(j,i);
for(int d=0;d<11;d++)
{
if(j==a[d]&&i==a[d+1])
break;
else
{
if(x>0&&d==10)
p++;
if(p%2==0)
putpixel(j,i,4);
}
}
}
}
```

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

OUTPUT

