

The page features a decorative design with three blue circles of varying sizes, each composed of concentric circles in different shades of blue. These circles are positioned in the top right, middle right, and bottom right areas. Thin blue lines intersect at the top left and extend towards the circles.

DRONACHARYA
College of Engineering

LAB MANUAL OF COMPUTER GRAPHICS

BY:-

Mrs. Manisha Saini

INTRODUCTION TO COMPUTER GRAPHICS

Computer Graphics is the pictorial representation of information using a computer program. In Computer Graphics, pictures as graphical subjects are represented as a collection of discrete picture element called pixels. Pixel is smallest addressable screen element.

1960s:

The advance in computer graphics was to come from one MIT student, Ivan Sutherland. In 1961 Sutherland created another computer drawing program called Sketchpad. Using a light pen, Sketchpad allowed one to draw simple shapes on the computer screen, save them and even recall them later. The light pen itself had a small photoelectric cell in its tip. Also in 1961 another student at MIT, Steve Russell, created the first video game, Spacewar. Written for the DEC PDP-1, Spacewar was an instant success and copies started flowing to other PDP-1 owners and eventually even DEC got a copy.

1970s:

The 1970s saw the introduction of computer graphics in the world of television. Computer Image Corporation (CIC) developed complex hardware and software systems such as ANIMAC, SCANIMATE and CAESAR. All of these systems worked by scanning in existing artwork, then manipulating it, making it squash, stretch, spin, fly around the screen, etc. . . Bell Telephone and CBS Sports were among the many who made use of the new computer graphics.

1980s:

During the early 1980's SIGGRAPH was starting to really take off. Catmull explains, "SIGGRAPH was a very good organization. It was fortuitous to have the right people doing the right things at the right time. It became one of the very best organizations where there is a lot of sharing and a lot of openness. Over the years it generated a tremendous amount of excitement and it was a way of getting a whole group of people to work together and share information, and it is still that way today."

1990s:

In May of 1990, Microsoft shipped Windows 3.0. It followed a GUI structure similar to the AppleMacintosh, and laid the foundation for a future growth in multimedia. While in 1990 only two of the nation's top ten programs ran under Windows, this rose to nine out of ten just a year later in 1991.

2000s:

The year 2000 really was 'the year of nVidia'. In december, nVidia acquired core assets of the once mighty 3DFX. This was a good reminder to all of us how quickly things can change in the industry.

2001s:

2001 saw a continuation of nVidia's dominance of the computer graphics market with an occasional competing product from ATI.
The big event of 2001 was probably be Microsoft's Xbox console.

2003s:

At the end of 2002 everyone was waiting for nVidia to launch their latest graphics chip (GeForce FX, alias nv30).

2004s:

2004 was a great year for graphics in computer games. Many of the titles that were expected in 2003 actually shipped in 2004. And just as many of us knew, a couple of games in particular raised the bar for graphical quality we expect from video games.

2005s:

The next generation of video consoles including Playstation 3 and the successor to Nintendo's GameCube (Probably going to be called *Revolution*) will be released in 2006.

APPLICATION/STRENGTH OF COMPUTER GRAPHICS

The following are also considered *graphics applications* :

- Paint programs : Allow you to create rough freehand drawings. The images are stored as bit maps and can easily be edited.
- Illustration/design programs: Supports more advanced features than paint programs, particularly for drawing curved lines. The images are usually stored in vector-based formats. Illustration/design programs are often called *draw programs*.
- Presentation graphics software: Lets you create bar charts, pie charts, graphics, and other types of images for slide shows and reports. The charts can be based on data imported from spreadsheet applications.
- Animation software: Enables you to chain and sequence a series of images to simulate movement. Each image is like a frame in a movie.
- CAD software: Enables architects and engineers to draft designs.
- Desktop publishing : Provides a full set of word-processing features as well as fine control over placement of text and graphics, so that you can create newsletters, advertisements, books, and other types of documents.
- Cartography - drawing maps
- Weather Maps - real time mapping, symbolic representations
- Satellite Imaging - geodesic images
- Photo Enhancement - sharpening blurred photos
- Medical imaging - MRIs, CAT scans, etc. - non-invasive internal examination.
- Engineering drawings - mechanical, electrical, civil, etc. - replacing the blueprints of the past.
- Typography - the use of character images in publishing - replacing the hard type of the past.
- Architecture - construction plans, exterior sketches - replacing the blueprints and hand drawings of the past.
- Art - computers provide a new medium for artists.

LIST OF EXPERMENTS

S.No.	AIM OF EXPERMENTS
1.	WAP TO DRAW A LINE USING DDA ALGORITHM.
2.	WAP TO DRAW A LINE USING BRESENHAM'S ALGORITHM.
3.	WAP TO DRAW A CIRCLE USING BRESENHAM'S CIRCLE ALGORITHM.
4.	WAP FOR BOUNDRY FILL ALGORITHM.
5.	WAP FOR FLOOD FIL ALGORITHM.
6.	WAP TO DRAW CIRCLE USING MID POINT ALGORITHM.
7.	WAP SHOWING TRANSLATION OF A TRIANGLE.
8.	WAP SHOWING SCALING OF A TRIANGLE.
9.	WAP SHOWING TRANSLATION OF A CIRCLE.
10.	WAP SHOWING SCALING OF CIRCLE.
11.	WAP SHOWING ROTATION OF A LINE.
12.	WAP SHOWING ROTATION OF A TRIANGLE.
13.	WAP TO MAKE A HUT,KITE AND A FISH.
14.	WAP TO ANIMATING A FISH.

Program No. : 1

Write a Program to draw a line using DDA Algorithm.

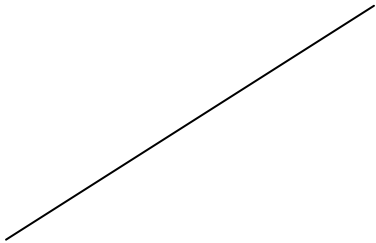
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gdriver=DETECT,gmode;
initgraph(&gdriver,&gmode,"c:\tc\bgi");
int x,y,x1,x2,y1,y2,dx,dy,step,m, xinc,yinc;
x=x1;
y=y1;
printf("Enter Ist end point of line");
scanf("%d%d",&x1,&y1);
printf("Enter 2nd end point of line");
scanf("%d%d",&x2,&y2);
dx=x2-x1;
dy=y2-y1;
m=(y2-y1)/(x2-x1);
if(abs(dx)>abs(dy))
step=abs(dx);
else
step=abs(dy);
xinc=dx/float(steps);
yinc=dy/float(steps);
putpixel(x,y,2);
for(k=0;k<=step;k++)
{
x=x+xinc;
y=y+yinc;
putpixel(x,y,2);
}
getch();
closegraph();
```

}

Output

Enter 1st end point of line: 100,200

Enter 2nd end point of line: 200,300



Program No. : 2

Write a Program to draw a line using Bresenham's Algorithm

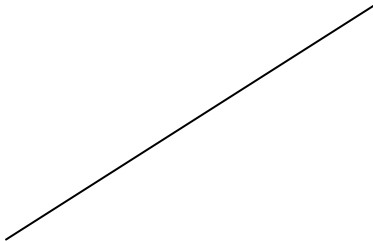
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x,y,x1,x2,y1,y2,dx,dy,p;
int gdriver=DETECT,gmode;
initgraph(&gdriver,&gmode,"c:\tc\bgi");
printf("Enter 1st end point of line");
scanf("%d%d",&x1,&y1);
printf("Enter 2nd end point of line");
scanf("%d%d",&x2,&y2);
x=x1;
y=y1;
putpixel(x,y,4);
dx=x2-x1;
dy=y2-y1;
p=2*dy-dx;
while(x<x2)
{
if(p<0)
{
x+=1;
p+=2*dy;
}
else
{
x+=1;
y+=1;
p+=2*dy-2*dx;
}
```



```
        }  
    putpixel(x,y,4);  
    }  
getch();  
closegraph();  
}
```

Output

Enter 1st end point of line: 100,200
Enter 2nd end point of line: 200,300



PROGRAM -3

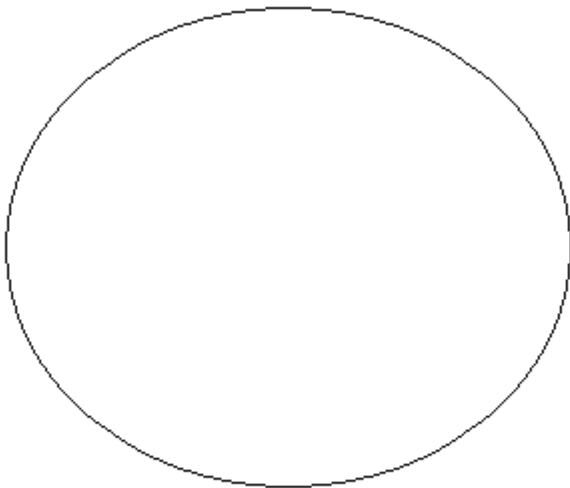
WRITE A PROGRAM FOR BRESENHAM CIRCLE ALGORITHM

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
int gd=DETECT,gm;
int r,x,y,p,xc=320,yc=420;
initgraph(&gd,&gm,"C://tc//bgi");
cleardevice();
printf("enter radius");
scanf("%d",&r);
x=0;
y=r;
putpixel(xc+x,yc-y,1);
p=3-(2*r);
for(x=0;x<=y;x++)
{
if(p<0)
{
y=y;
p=(p+4*x)+6;
}
else
{
y=y-1;
p=p+((4*(x-y)+10));
}
putpixel(xc+x,yc-y,1);
putpixel(xc-x,yc-y,2);
putpixel(xc+x,yc+y,3);
putpixel(xc-x,yc+y,4);
```

```
putpixel(xc+y,yc-x,5);  
putpixel(xc-y,yc-x,6);  
putpixel(xc+y,yc+x,7);  
putpixel(xc-y,yc+x,8);  
}  
getch();  
closegraph();  
}
```

OUTPUT

Enter radius 100



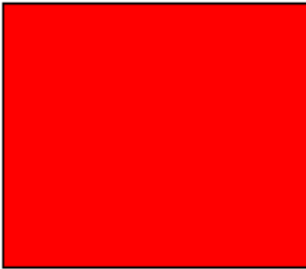
PROGRAM -4

WRITE A PROGRAM FOR BOUNDARY FILL ALGORITHM

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void boundaryFill(int x,int y)
{
int interiorColor;
interiorColor=getpixel(x,y);
if((interiorColor !=WHITE)&&(interiorColor !=RED))
{
putpixel(x,y,RED);
boundaryFill(x+1,y);
boundaryFill(x,y+1);
boundaryFill(x-1,y);
boundaryFill(x,y-1);
}
delay(1);
}
void main()
{
void boundaryFill(int,int);
int x,y,n,i;
int gd=DETECT,gm;
clrscr();
initgraph(&gd,&gm,"C:\\TC\\BGI");
line(50,50,100,50);
line(100,50,100,100);
line(100,100,50,100);
line(50,100,50,50);
x=78,y=78;
```

```
boundaryFill(x,y);  
getch();  
closegraph();  
}
```

OUTPUT



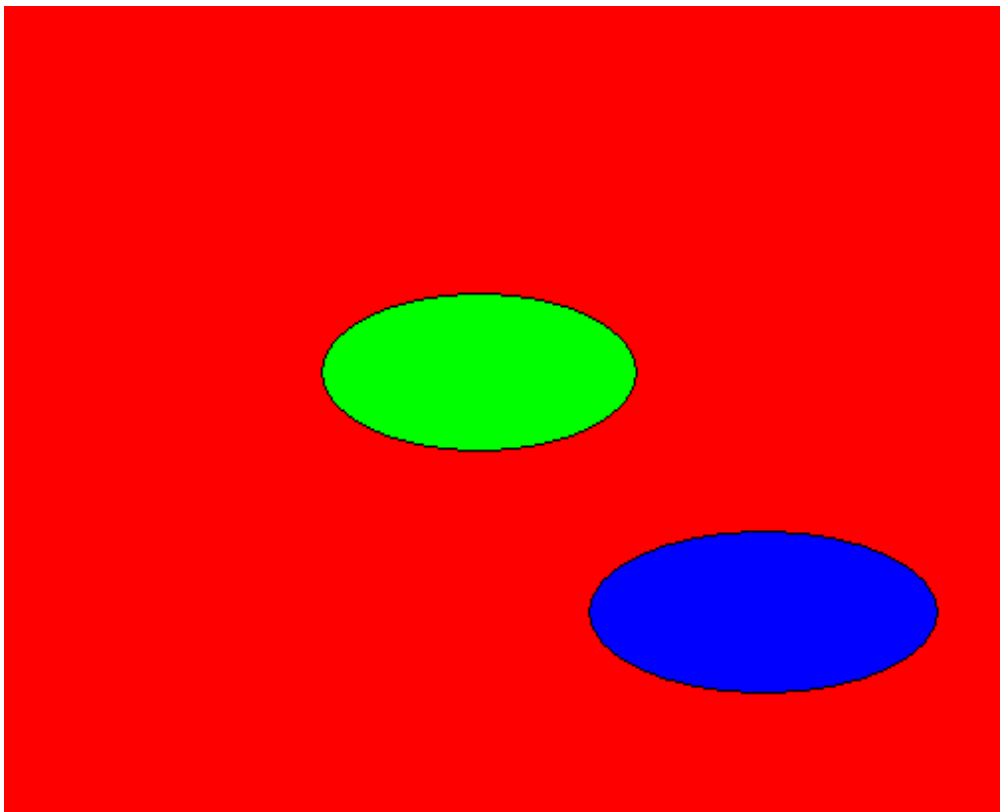
PROGRAM -5
WRITE A PROGRAM FOR FLOOD FILL ALGORITHM

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void floodFill(int x,int y,int ncolor,int ocolor)
{
if(getpixel(x,y)==ocolor)
{
putpixel(x,y,ncolor);
floodFill(x+1,y,ncolor,ocolor);
floodFill(x-1,y,ncolor,ocolor);
floodFill(x,y+1,ncolor,ocolor);
floodFill(x,y-1,ncolor,ocolor);
}
delay(1);
}
void main()
{
int x,y,ncolor=BLUE,ocolor=WHITE;
int midx,midy;
int gd=DETECT,gm;
initgraph(&gd,&gm,"C://tc//bgi");
cleardevice();
printf("enter seed point:");
scanf("%d%d",&x,&y);
midx=getmaxx()/2;
midy=getmaxy()/2;
setbkcolor(RED);
setpalette(ocolor,GREEN);
fillellipse(midx,midy,50,25);
fillellipse(midx+100,midy+100,50,25);
```

```
floodFill(x,y,ncolor,ocolor);  
getch();  
closegraph();  
}
```

OUTPUT

enter seed point:
320
240



PROGRAM-6

WRITE A PROGRAM FOR MID POINT CIRCLE ALGORITHM

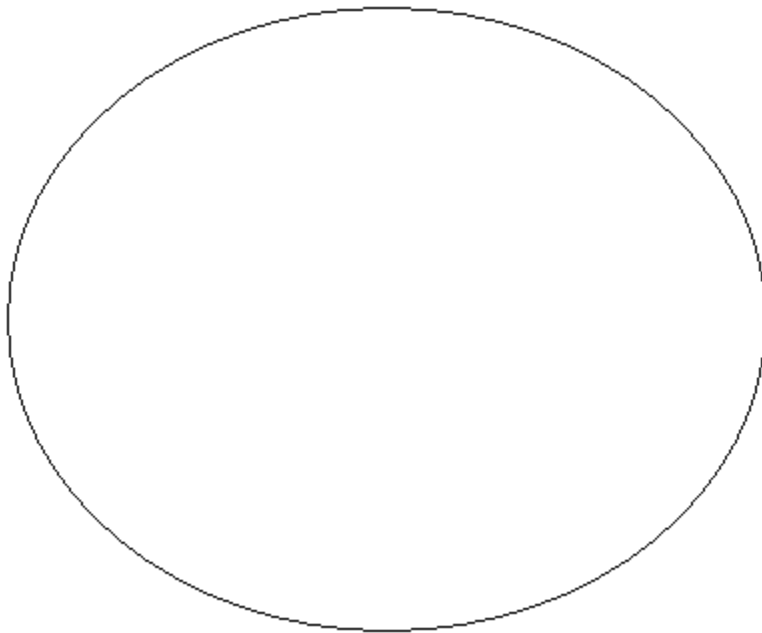
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
int gd=DETECT,gm;
int r,x,y,p,xc=320,yc=420;
initgraph(&gd,&gm,"C://tc//bgi");
cleardevice();
printf("enter radius");
scanf("%d",&r);
x=0;
y=r;
putpixel(xc+x,yc-y,1);
p=1-r;
for(x=0;x<=y;x++)
{
if(p<0)
{
y=y;
p=(p+2*x)+3;
}
else
{
y=y-1;
p=p+((2*(x-y)+5));
}
putpixel(xc+x,yc-y,1);
putpixel(xc-x,yc-y,2);
putpixel(xc+x,yc+y,3);
putpixel(xc-x,yc+y,4);
}
```



```
putpixel(xc+y,yc-x,5);  
putpixel(xc-y,yc-x,6);  
putpixel(xc+y,yc+x,7);  
putpixel(xc-y,yc+x,8);  
}  
getch();  
closegraph();  
}
```

OUTPUT

enter radius 200



PROGRAM -7
WRITE A PROGRAM FOR TRANSLATION OF A TRIANGLE

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
int x1,y1,x2,y2,x3,y3,tx,ty;
int gd=DETECT,gm;
initgraph(&gd,&gm,"C://tc//bgi");
printf("enter x1 and y1");
scanf("%d%d",&x1,&y1);
printf("enter x2 and y2");
scanf("%d%d",&x2,&y2);
printf("enter x3 and y3");
scanf("%d%d",&x3,&y3);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x1,y1,x3,y3);
printf("enter translation vector along x-axis");
scanf("%d",&tx);
printf("enter translation vector along y-axis");
scanf("%d",&ty);
line(x1+tx,y1+ty,x2+tx,y2+ty);
line(x2+tx,y2+ty,x3+tx,y3+ty);
line(x1+tx,y1+ty,x3+tx,y3+ty);
getch();
closegraph();
}
```

OUTPUT

enter x1 and y1

100

250

enter x2and y2

200

250

enter x3 and y3

150

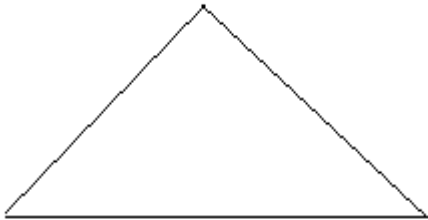
150

enter translation vector along x-axis

100

enter translation vector along y-axis

100



PROGRAM: 8

Write A Program For Scaling of a Triangle

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

void main()
{
int c,x1,y1,x2,y2,x3,y3,sx,sy;
int gd=DETECT,gm;
initgraph(&gd,&gm," ");
printf("Enter coord of 1st pixel");
scanf("%d%d",&x1,&y1);
printf("Enter coordinate of 2nd pixel");
scanf("%d%d",&x2,&y2);
printf("Enter coordinate of 3rd pixel");
scanf("%d%d",&x3,&y3);
line(x1,y1,x2,y2);
line(x1,y1,x3,y3);
line(x3,y3,x2,y2);
printf("Enter the scaling vector along x-axis");
scanf("%d",&sx);
printf("Enter the scaling vector along y-axis");
scanf("%d",&sy);
i:
printf("Enter your choice \n Increment \n Decrement");
scanf("%d",&c);
switch(c)
{
case 1:
line(x1*sx,y1*sy,x2*sx,y2*sy);
line(x1*sx,y1*sy,x3*sx,y3*sy);
line(x3*sx,y3*sy,x2*sx,y2*sy);
case 2:
line(x1/sx,y1/sy,x2/sx,y2/sy);
line(x1/sx,y1/sy,x3/sx,y3/sy);
line(x3/sx,y3/sy,x2/sx,y2/sy);
```

```
break;
default:
printf("You have entered a wrong choice");
goto i;
}
getch();
closegraph();
}
```

OUTPUT

Enter coordinates of 1st pixel: 100 100

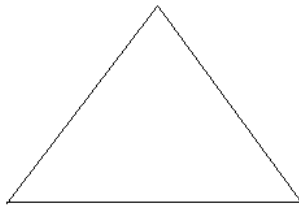
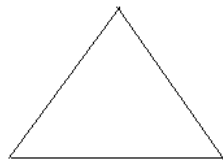
Enter coordinates of 2nd pixel: 50 150

Enter coordinates of third pixel: 150 150

Enter scaling vector(Sx,Sy):

Enter your choice

1. Increment
2. Decrement



PROGRAM: 09

Write A Program For Translation Of A Circle

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
void main()
{
int x,y,r,tx,ty ;
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\tc");
printf("enter center coordinates of circle");
scanf("%d%d",&x,&y);
printf("enter radius of circle");
scanf("%d",&r);
circle(x,y,r);
printf("enter translation vector along x-axis");
scanf ("%d",&tx);
printf("enter translation vector along y-axis");
scanf ("%d",&ty);

circle(x+tx,y+ty,r);
getch();
closegraph();
}
```

OUTPUT

Enter center coordinates of circle

300

300

Enter the radius of circle

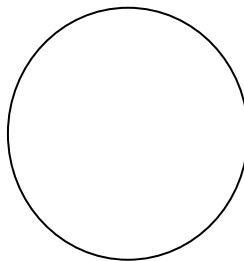
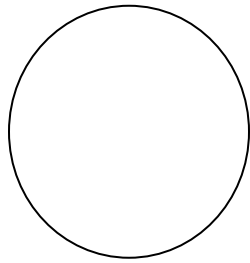
30

Enter translation vector along x-axis

100

Enter translation vector along y-axis

100



PROGRAM: 10

Write a Program for scaling in a circle

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

void main()
{
int s,r,c,x,y;
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\tc");
printf("Enter centre coordinates of circle: ");
scanf("%d%d",&x,&y);
printf("Enter radius");
scanf("%d",&r);
circle(x,y,r);
printf("Enter the scaling vector: ");
scanf("%d",&s);
i:
printf("Enter your choice \n 1.Increment \n2. Decrement");
scanf("%d",&c);
switch(c)
{
case 1:
circle(x,y,r*s);
break;
case 2:
circle(x,y,r/s);
break;
default:
printf("You have entered a wrong choice");
goto i;
}
getch();
closegraph();
}
```


OUTPUT

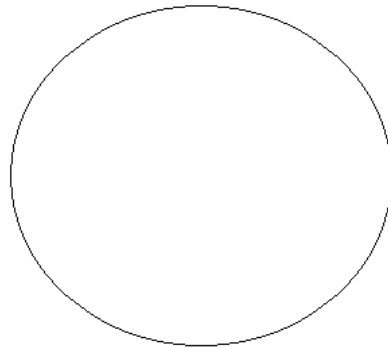
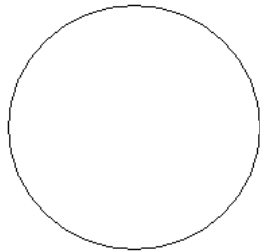
Enter coordinates of circle: 300 300

Enter radius of circle: 100

Enter scaling vectors: 2

Enter your choice

1. Increment
2. Decrement



Program-11
Program to implement rotation of line

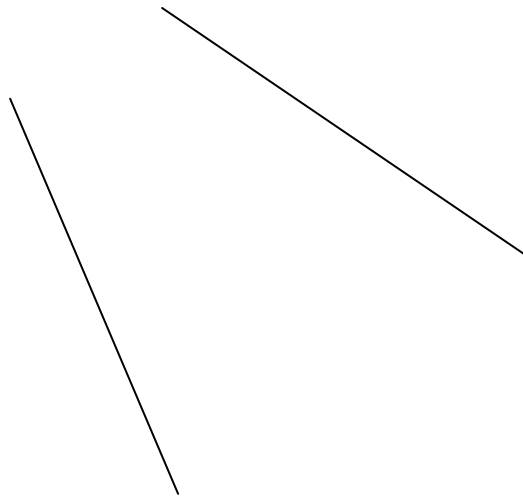
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
int gdriver=DETECT,gmode;
float theta;
int x1,y1,x2,y2,x11,y11,x22,y22;
initgraph(&gdriver,&gmode,"c://tc//bgi");
printf("enter the coordinate of 1st point");
scanf("%d%d",&x1,&y1);
printf("enter the coordinate of 2nd point");
scanf("%d%d",&x2,&y2);
line(x1,y1,x2,y2);
getch();
printf("enter the rotation angle");
scanf("%f",&theta);
theta=theta*3.14/180;
x11=(x1*cos(theta))-(y1*sin(theta));
y11=(x1*sin(theta))+(y1*cos(theta));
x22=(x2*cos(theta))-(y2*sin(theta));
y22=(x2*sin(theta))+(y2*cos(theta));
line(x11,y11,x22,y22);
getch();
closegraph();
}
```

OUTPUT

Enter the first co-ordinate 100 ,
100

Enter the second co-ordinate 200
200

Enter the rotation angle 45



Program-12
Program to implement rotation of triangle

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
int gd driver=DETECT,gmode;
float theta;
float x1,y1,x2,y2,x3,y3,x11,y11,x22,y22,x33,y33;
initgraph(&gdriver,&gmode,"c://tc//bgi");
printf("enter the 1st coordinate point of a triangle");
scanf("%d%d",&x1,&y1);
printf("enter the 2nd coordinate point of triangle");
scanf("%d%d",&x2,&y2);
printf("enter the 3rd coordinate point of triangle");
scanf("%d%d",&x3,&y3);
line(x1,y1,x2,y2);
line(x1,y1,x3,y3);
line(x3,y3,x2,y2);
getch();
printf("enter the rotation angle");
scanf("%f",&theta);
theta=theta*3.14/180;
x11=(x1*cos(theta))-(y1*sin(theta));
y11=(x1*sin(theta)+(y1*cos(theta));
x22=(x2*cos(theta))-(y2*sin(theta));
y22=(x2*sin(theta)+(y2*cos(theta));
x33=(x3*cos(theta))-(y3*sin(theta));
y33=(x3*sin(theta)+(y3*cos(theta));
line(x11,y11,x22,y22);
line(x11,y11,x33,y33);
line(x33,y33,x22,y22);
getch();
closegraph();
```

Output

Enter the 1st co-ordinate point of triangle 50

100

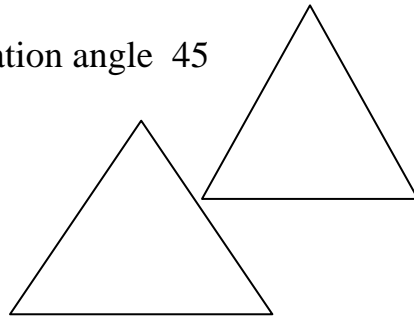
Enter the 2nd co-ordinate point of triangle 100

150

Enter the 3rd co-ordinate point of triangle 200

150

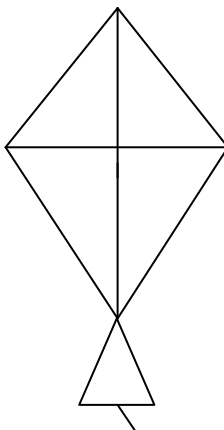
Enter the rotation angle 45



Program-13
Write a program to make a kite

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int a,b,gd =DETECT,gm,i;
clrscr();
init graph(&gd,&gm,"c:\tc\bgi ");
line(100,100,50,180);
line(100,100,150,180);
line(50,180,100,250);
line(150,180,100,250);
line(100,100,100,250);
line(50,180,150,180);
line(100,250,70,300);
line(100,250,130,300);
line(70,300,130,300);
line(100,300,120,320);
line(120,320,80,340);
line(80,340,120,360);
line(120,360,80,380);
set color(4);
getch();
closegraph();
}
```

OUTPUT



Program :14
Write a program to make a fish

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd =DETECT,gm;
clrscr();
init graph(&gd,&gm,"c:\tc\bgi");
set color(6);
ellipse(200,200,0,360,50,30);
line(250,200,280,170);
line(280,170,280,230);
line(280,230,250,200);
circle(160,190,3);
getch();
closegraph();
}
```

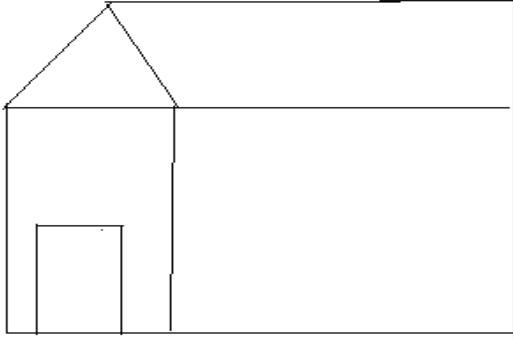
OUTPUT



Program :15
Write a program to make a hut

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd =DETECT,gm;
clrscr();
init_graph(&gd,&gm,"c:\tc\bgi");
set_color(6);
rectangle(50,180,150,300);
rectangle(150,180,320,300);
rectangle(80,250,120,300);
line(100,100,50,180,);
line(100,100,300,100);
line(100,100,300,100);
line(300,100,320,180);
getch();
closegraph();
}
```

OUTPUT



PROGRAM-16

ANIMATION OF FISH PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm,i;
initgraph(&gd,&gm,"c:/tc/bgi");
clrscr();
setcolor(2);
for(i=500;i>=0;i--)
{
clearviewport();
ellipse(200+i,200,0,360,50,30);
line(250+i,200,280+i,170);
line(280+i,170,280+i,230);
line(280+i,230,250+i,200);
circle(160+i,190,3);
}
setcolor(1);

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

Output

