S.No	Program	Page No	T.S
1.	WAP to use putpixel function	5	
2.	WAP to draw a line using line function	7	
3	WAP to draw a rectangle using line function	9	
4.	WAP to draw triangle using line function.	11.	
5.	WAP to draw square using rectangle function	13.	
6.	WAP to draw rectangle using rectangle function.	15.	
7.	WAP to draw hut using line fun^n	17.	
8.	WAP to draw circle using circle fun^n	19	
9.	WAP to draw softy	21.	
10.	WAP to draw arc using arc fun^n	23.	
11.	WAP to draw ellipse using ellipse fun^n	25	
12.	WAP to draw a star using line function	27	
13.	WAP to draw rectangle using lineto fun^n	29	
14.	WAP to use moveto function	31	
15.	WAP to use outtext function	33	
16.	WAP to use outtextxy function	35	
17.	WAP to use bar function	37	
18.	WAP to use 3dbar function	39	
19.	WAP to use fillellipse function	41	
20.	WAP to use setbkcolor function	43	
21.	WAP to use setcolor function	45	
22.	WAP to use setfillstyle() or floodfill() function	47	
23	WAP to draw happy and sad face	49	

S.No	Program	Page No.	T.S.
24.	WAP to draw Traffic light	51	
25.	WAP to use getmaxx() and getmaxy()	53.	
26.	WAP to draw name using line function	55	
27.	WAP to draw cap	57	
28	WAP to draw clock	59	
29.	WAP to draw bar graph using bar() function	61	
30.	WAP to draw 3dbar graph using bar3d() function	63	
31.	WAP to use fillpoly()	65	
32.	WAP to draw piechart	67	
33.	WAP to use pieslice	69	
34	WAP to generate circle on screen randomly	71	
35.	WAP to draw computer	73	
36.	WAP to draw fish	77	
37.	WAP to draw triangle using drawpoly() func^n	79	
38.	WAP to draw pot	81	
39	WAP to draw flower pot	83	
40	WAP to draw digital clock	87.	
41.	WAP to draw bouncing ball	89	
42.	WAP to draw TURBOC3 screen	91	
43	WAP to draw solar system	93	
44.	WAP to draw Night Sky	97	
45.	WAP to draw lock and key	99	
46.	WAP to draw pencil	101	
47	WAP to draw chess Board	103	
48	WAP to draw Flying Bird	107	
49	WAP to draw a line dynamically	111	
50	WAP to draw Indian flag	113	
51	WAP to draw rainy day umbrella	115	

S.No	Programs	Page No.	T.S
52.	WAP to draw man object	119	
53.	WAP to draw cos waves	121	
54.	WAP to draw Kite	123	
55	WAP to move any object	125	
56.	WAP to draw olympic ring	127	
57.	WAP to implement DDA line Algo.	129	
58.	WAP to implement Bresenham's line Algorithm	131	
59.	WAP to implement mid point circle Algorithm	135	
60.	WAP to implement Bresenham's circle Algorithm	139	
61.	WAP to draw rainbow	143	
62.	WAP to draw a moving cycle	145	
63	WAP to use different textstyle and color	149	
64.	WAP to draw football ground	151	
65	WAP to illustrate different linestyle	153	
66	WAP to draw moving car	155	
67.	WAP to draw cricket pitch	159	
68	WAP to draw cartoon character	163	

Computer Graphics

Computer graphics is an art of drawing pictures, lines, charts etc. Using computer with the help of programming computer graphics is made up of number of pixels.
Computer graphics is a use of hardware and software to create manipulate & present picture & image.

graphics.h

The graphics.h header file provide access to a simple graphics library that makes it possible to draw line, rectangle, ovals, arcs, polygons, image and strings.

initgraph

Initgraph is used to initialize the graphical system by loading graphics driver from disk and there by putting the system into graphics mode.

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

Graph Driver

Graph driver is the integer that which graphics driver is to be used. It is a pointer to an integer specifying the graphics driver to be used. In all our program use will DETECT macro of graphics.h library that instruct compiler for auto detection of graphics driver.

Graphics Mode

It is a pointer to an integer that specifies the graphics mode to be used. If * graphics mode gdriver is set to the highest resolution sets * gmode to the highest resolution available for the detected driver.

Driver Directory Path

That specifies the directory path where graphics driver files (BGI files) are located. If directory path is not provided then it will search for driver file in current working directory.

closegraph

closegraph de-allocate all memory allocated by the graphic system, then restore the screen to the mode it was in before you called initgraph.

output

Program - 01

// WAP to use putpixel function.

syntax :-

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

input :-

```
#include <graphics.h>
#include <conio.h>
void main()
{
    int gd = DETECT, gm;
    clrscr();
    initgraph(&gd, &gm, " ");
    putpixel(100, 200, RED);
    closegraph();
    getch();
}
```

output -

Program - 02

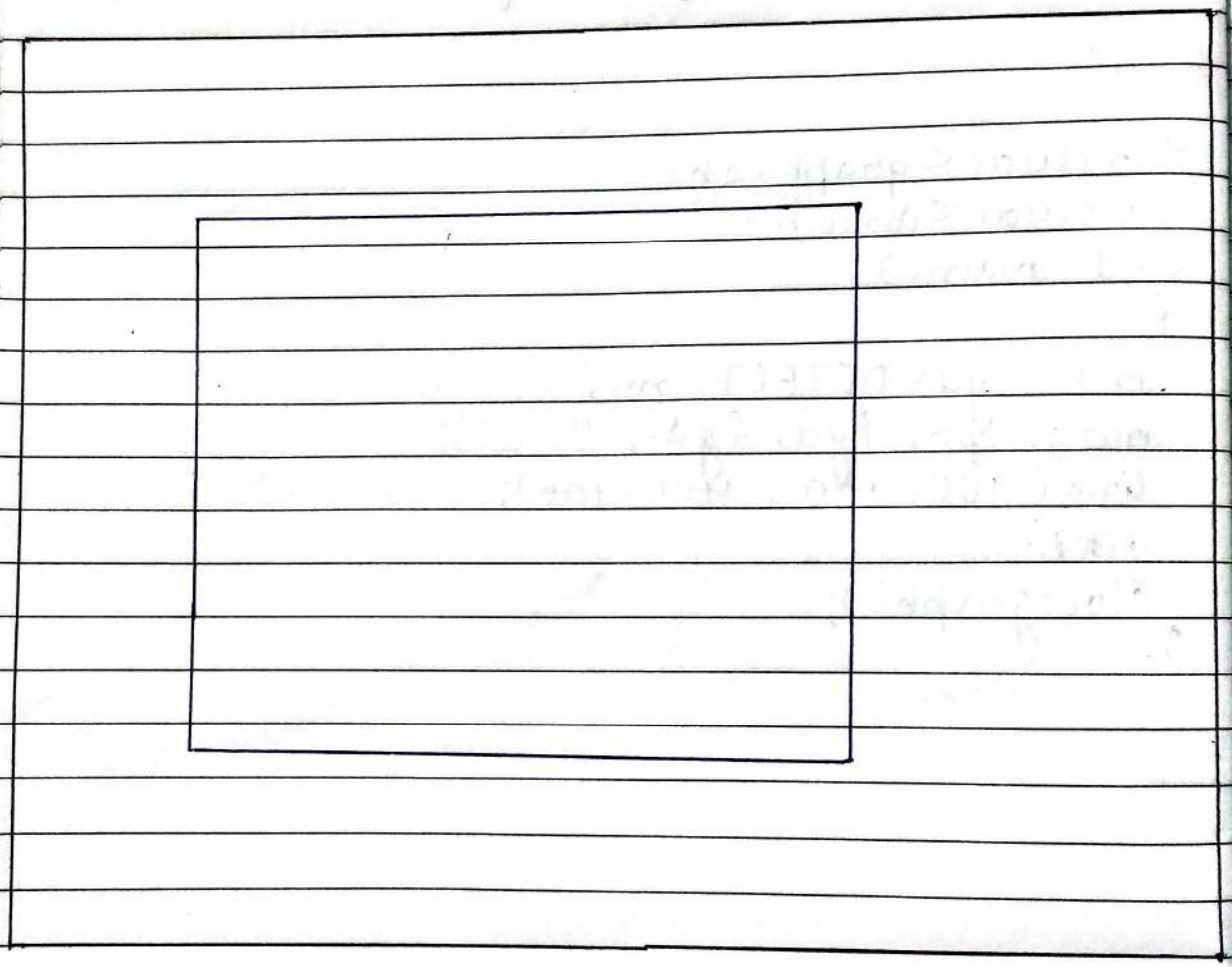
// WAP to draw a line using line function.

syntax :- line(x_1, y_1, x_2, y_2);

input :

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, "  ");
  line(100, 100, 400, 100);
  getch();
  closegraph();
}
```

output



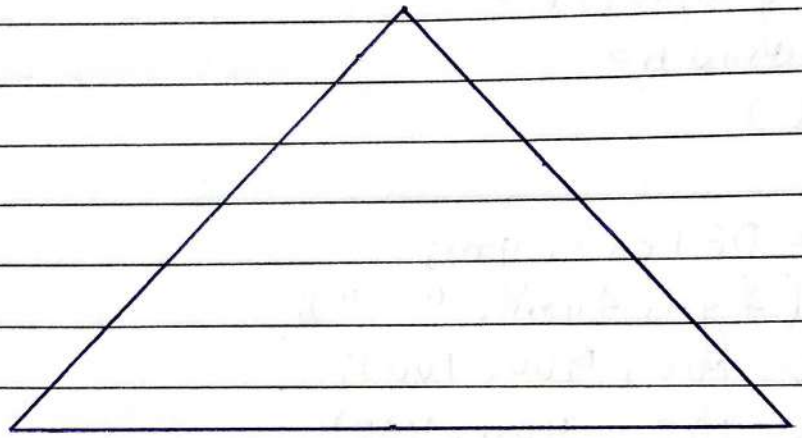
Program - 03

// WAP to draw a rectangle using line function.

input →

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd= DETECT, gm;
initgraph(&gd, &gm, " ");
line(100, 100, 400, 100);
line(400, 100, 400, 400);
line(400, 400, 100, 400);
line(100, 400, 100, 100);
getch();
closegraph();
}
```

output



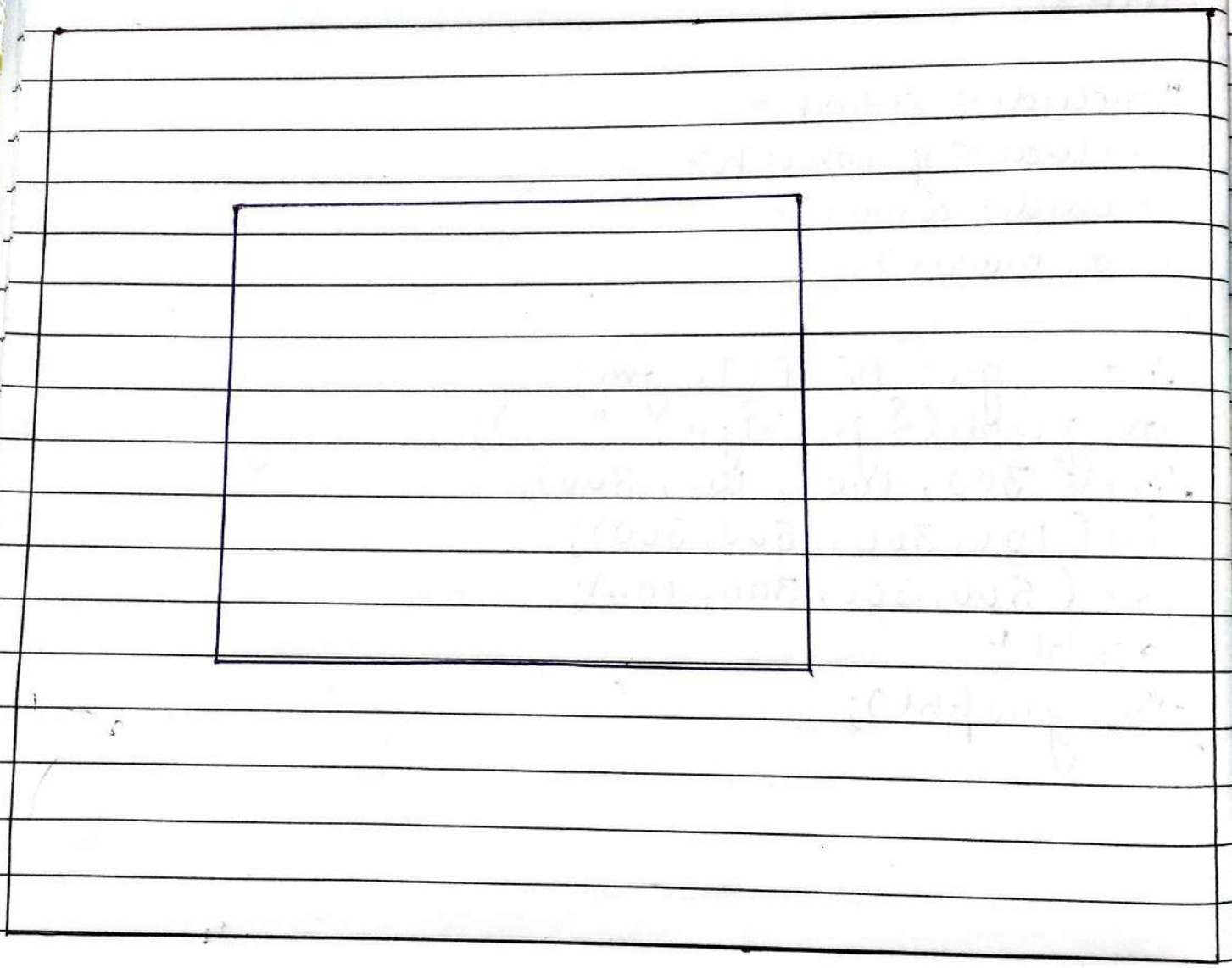
Program-04

// WAP to draw triangle using line function.

input->

```
#include< stdio.h>
#include <graphics.h>
#include<conio.h>
void main()
{
 int  gd= DETECT, gm;
 initgraph(&gd, &gm, " ");
line( 300, 100 , 100, 300);
line(100, 300, 500, 300);
line ( 500, 300, 300, 100);
 getch();
closegraph();
}
```

output



Program-05

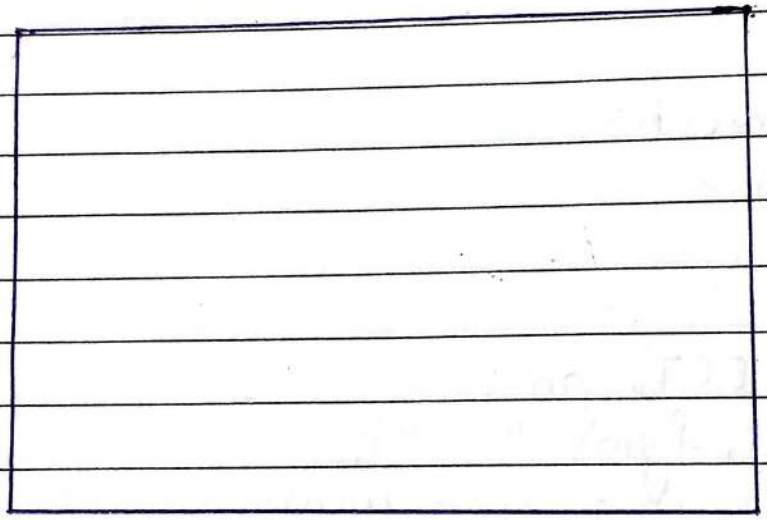
// WAP to draw square using rectangle function

Syntax- rectangle (a, a, b, b);

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd= DETECT, gm;
 initgraph(&gd, &gm, " ");
 rectangle (100, 100, 400, 400);
 getch();
 closegraph();
}
```

output



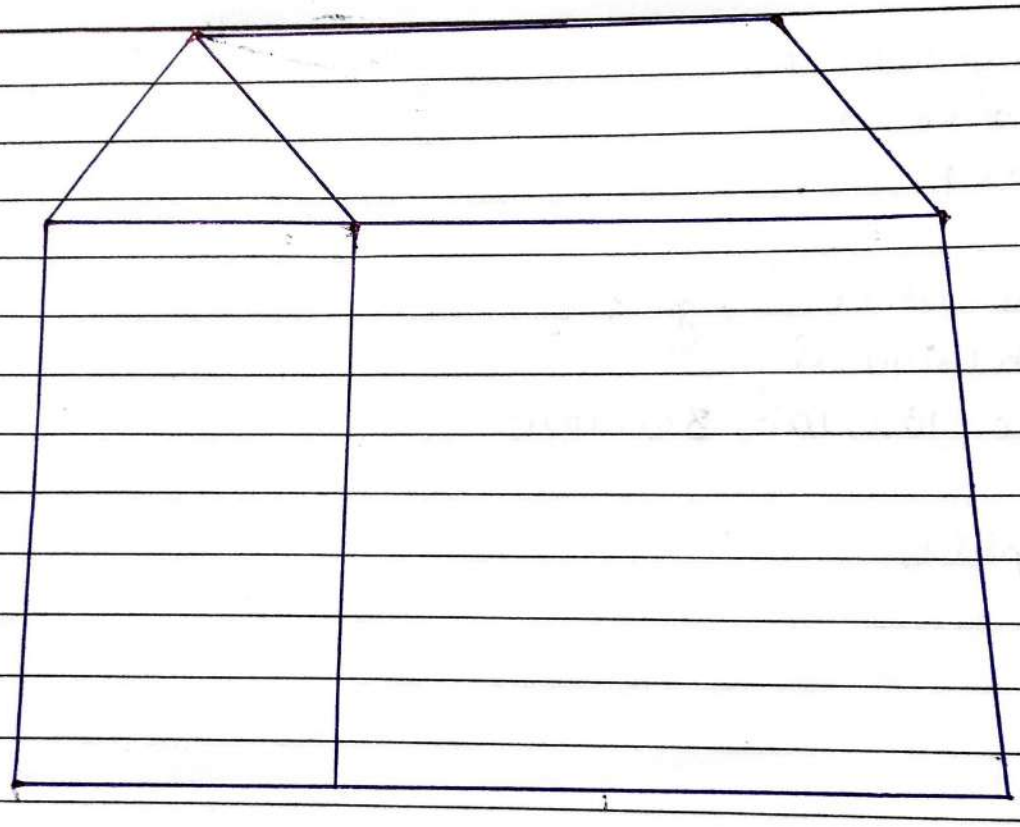
Program - 06

// WAP to draw rectangle using rectangle function.

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  rectangle(100, 100, 300, 400);
  getch();
  closegraph();
}
```

output-



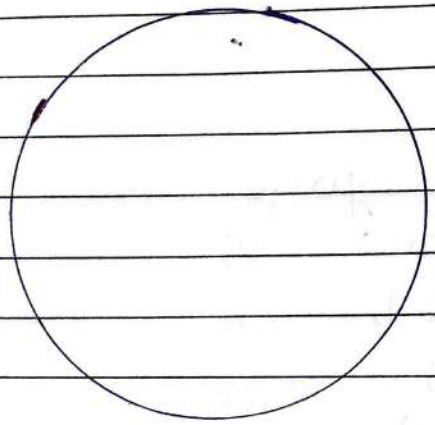
Program - 07

// WAP to draw hut using line function

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd= DETECT, gm;
 initgraph(&gd,&gm,"  ");
 line(50,0,350,0);
 line(0,100,50,0);
 line(0,100,100,100);
 line(100,100,50,0);
 line(100,100,400,100);
 line(350,0,400,100);
 line(0,100,0,400);
 line(100,400,100,400);
 line(100,400,400,400);
 line(400,400,400,400);
 getch();
 closegraph();
}
```

output-



Program-08

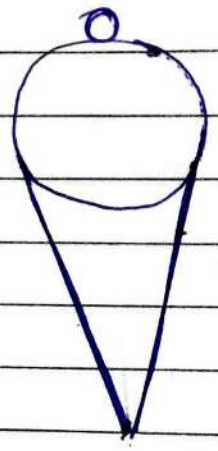
// WAP to draw circle using circle function

syntax- circle(x, y, radius);

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd=DETECT, gm;
 initgraph(&gd,&gm," ");
 circle(200, 200, 100);
 getch();
 closegraph();
}
```

output -



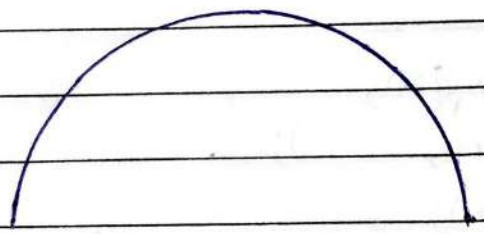
Program - 09

// WAP to draw softy

input:-

```
#include<graphics.h>
#include<conio.h>
void main( )
{
 int gd=DETECT, gm;
 initgraph(&gd,&gm, " ");
 circle(300, 100, 10);
 circle (300, 155, 45);
 line(300, 300, 262, 180);
 line(300,300,338, 180);
  getch( );
 closegraph( );
}
```

output -



Program - 10

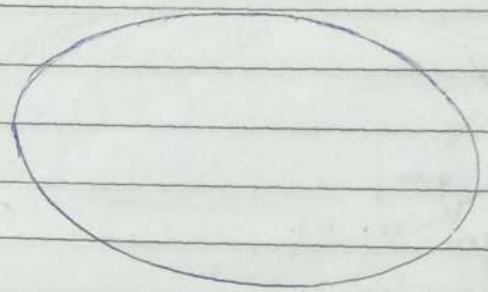
// WAP to draw arc using arc function

syntax - arc(x, y, S-A, E-A, radius);

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd= DETECT, gm;
 initgraph(&gd, &gm, " ");
 arc(200, 200, 0, 180, 100);
 getch();
 closegraph();
}
```

Output -



Program - 11

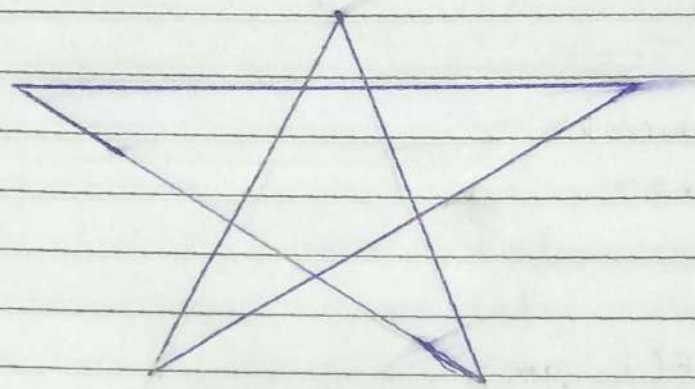
// WAP to draw ellipse using ellipse function

syntax- ellipse(x, y, S.A, E.A, r_1, r_2);

input-

```
#include <graphics.h>
#include <conio.h>
void main()
{
 int gd = DETECT, gm;
 initgraph(&gd, &gm," ");
 ellipse(300, 200, 0, 360, 200, 100);
 getch();
 closegraph();
}
```

Output-



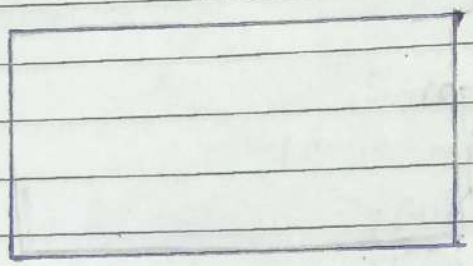
Program-12

//WAP to draw a star using line function

Input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd=DETECT, gm;
 initgraph(&gd, &gm, " ");
 line(200,50,140,200);
 line(200,50,260,200);
 line(140,200,280,120);
 line(260,200,120,120);
 line(120,120,280,120);
 getch();
 closegraph();
}
```

Output



Program-13

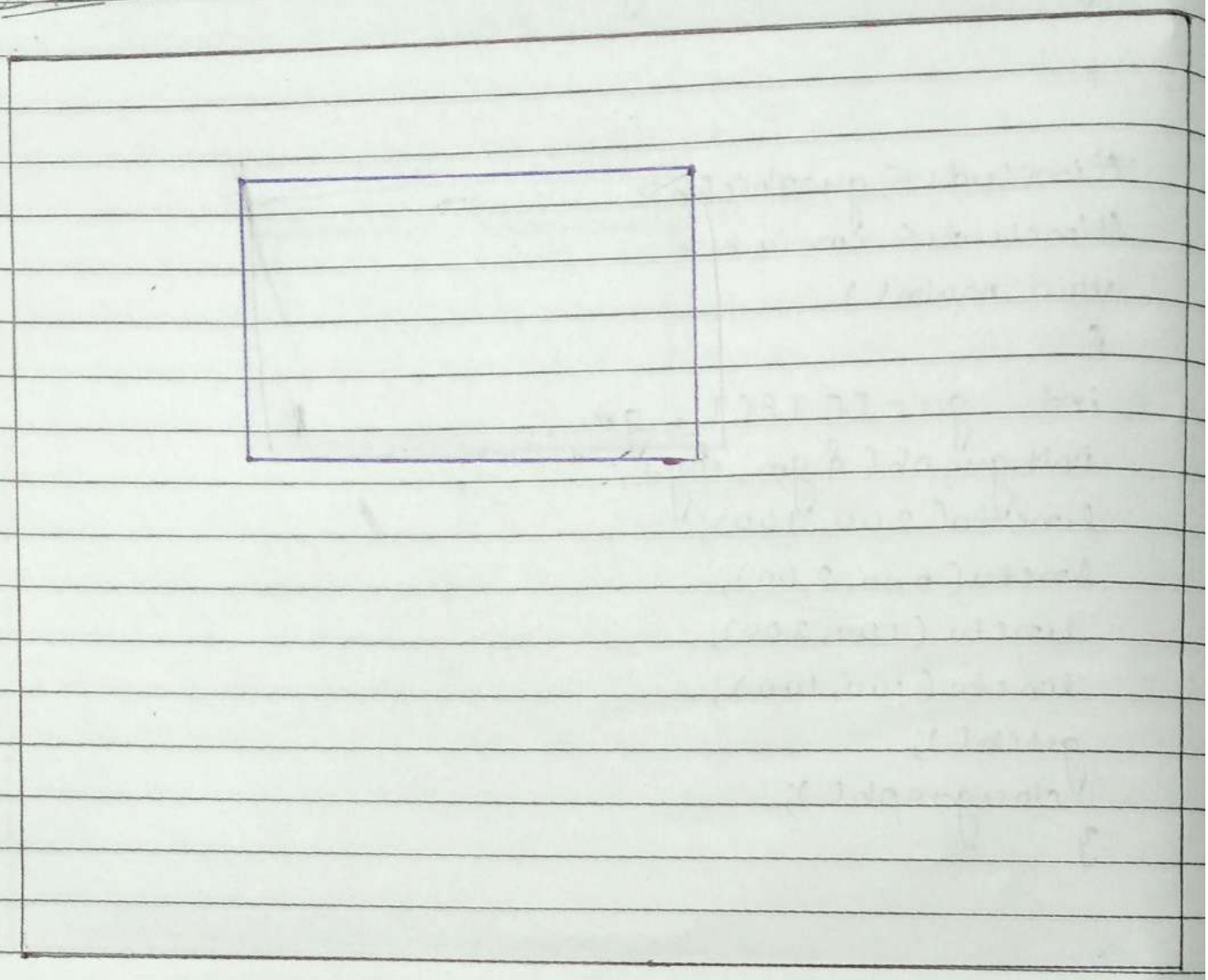
// WAP to draw rectangle using lineto function

Syntax- lineto(x,y);

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
int gd=DETECT, gm;
initgraph(&gm, &gd, " ");
lineto(200,100);
lineto(200,200);
lineto(100,200);
lineto(100,100);
getch();
closegraph();
}
```

output -



Program - 14

// WAP to use moveto function.

Syntax-

input-

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph( &gd, &gm, " ");
  moveto (100, 100);
  lineto( 200, 100);
  moveto( 200, 100);
  lineto( 200, 200);
  moveto( 200, 200);
  lineto( 100, 200);
  moveto( 200, 100);
  lineto( 100, 100);
  getch();
  closegraph();
}
```

output -

We are Coders

Program-15

// WAP to use outtext() function

Syntax- outtext("text");

input-

```
#include <graphics.h>
#include<conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  outtext("We are Coders");
  getch();
  closegraph();
}
```

Output -

Bhavy Sharma
Also known as
Coder Shab

Program-16

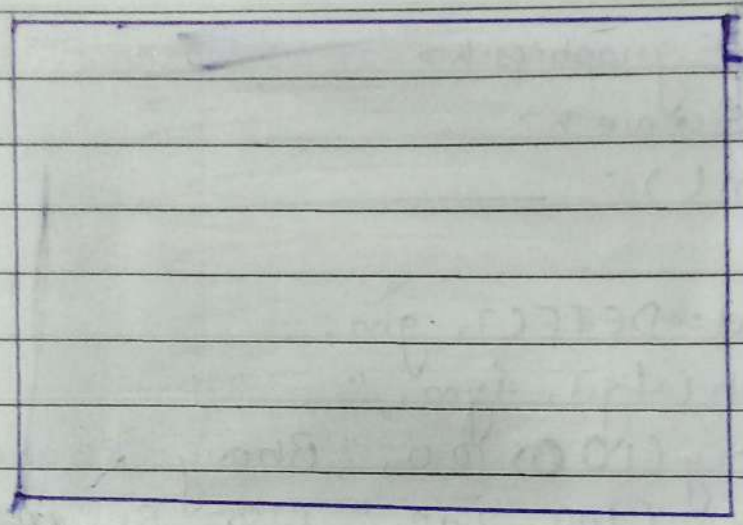
// WAP to use outtextxy( ) function

Syntax- outtextxy(x, y, "text");

input-

```
#include <graphics.h>
#include <conio.h>
void main( )
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, "  ");
  outtextxy(100, 100, "Bhavy Sharma");
  outtextxy(100, 200, "Also Known as");
  outtextxy(100, 300, "Coder Shab");
  getch( );
  closegraph( );
}
```

Output-



Program - 17

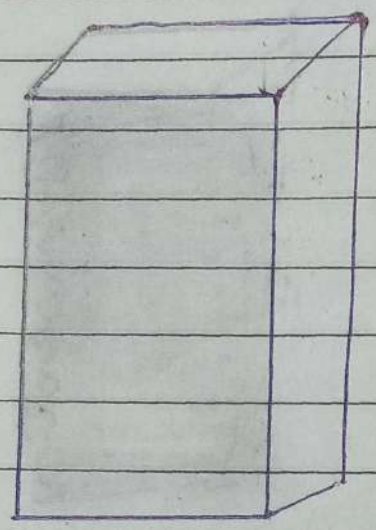
// WAP to use bar() function.

Syntax - bar(x_1, y_1, x_2, y_2);

input -

```
#include< graphics.h >
#include< conio.h >
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  bar( 100, 100, 300, 300);
  getch();
  closegraph();
}
```

output -



Program-18

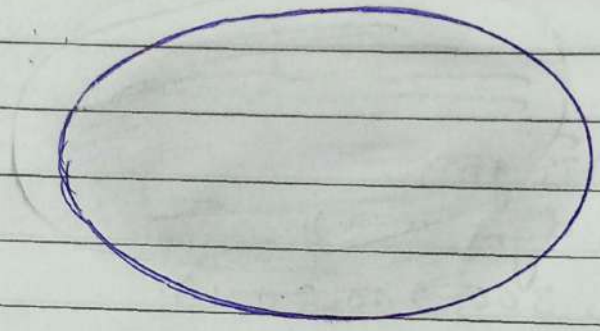
// WAP to use bar3d();

Syntax - bar3d(x_1, y_1, x_2, y_2, depth, topflag);

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  bar3d(100, 100, 300, 200, 20, 1);
  getch();
  closegraph();
}
```

Output -



Program - 19

// WAP to use fillellipse( ) function.

Syntax -

input -

```
#include <graphics.h>
#include<conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  fillellipse(300,200, 200, 100);
  getch();
  closegraph();
```

output-

Program - 20

// WAP to use setbkcolor() function.

Syntax: setbkcolor(COLOR);

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
 int gd = DETECT, gm;
 initgraph(&gd, &gm, " ");
 setbkcolor(RED);
 getch();
 closegraph();
}
```

output-

Program-21

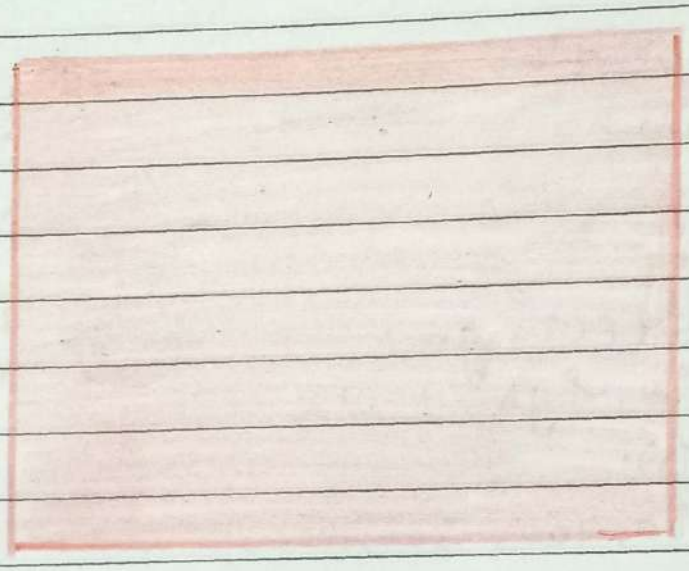
// WAP to use setcolor function.

Syntax- setcolor(COLOR);

input-

```
#include< graphics.h>
#include< conio.h>
void main( )
{
  int gd= DETECT, gm;
  initgraph( &gd, &gm, " ");
  setcolor( RED);
  rectangle( 100, 100, 300, 300);
  getch( );
  closegraph( );
}
```

Output-



Program - 22

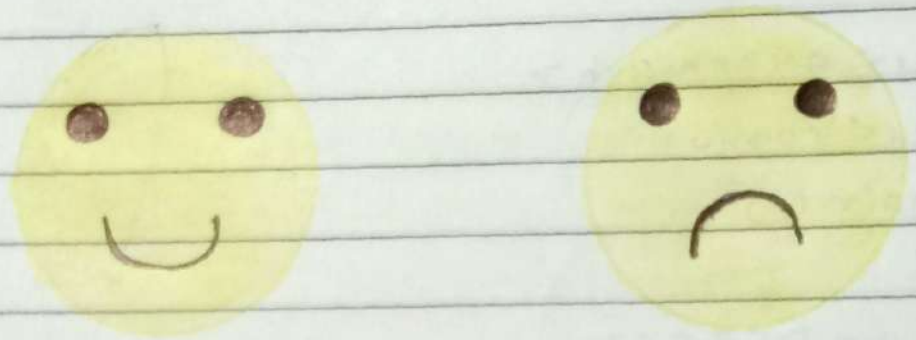
// WAP to use setfillstyle() or floodfill() function.

Syntax- setfillstyle(COLOR_PATTERN, COLOR);
floodfill(x, y, COLOR);

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  setcolor(RED);
  rectangle(100, 100, 300, 300);
  setfillstyle(SOLID_FILL, RED);
  floodfill(150, 150, RED);
  getch();
  closegraph();
}
```

Output -



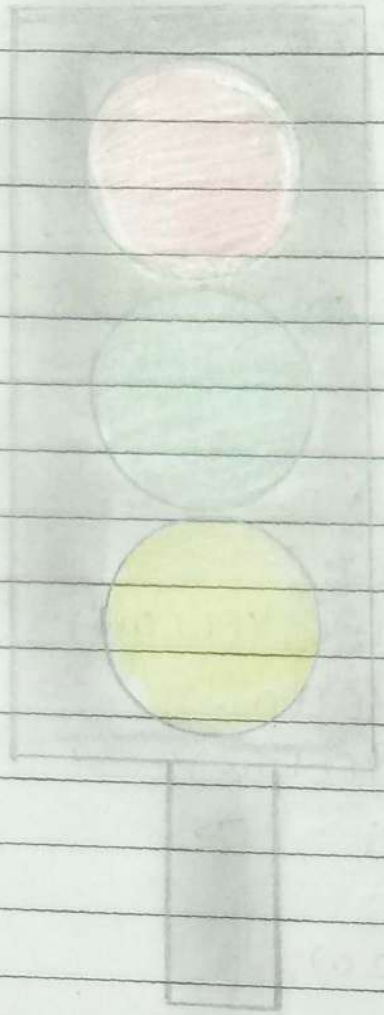
Program - 23

// WAP to draw happy and sad face.

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
int gd=DETECT, gm;
initgraph(&gd, &gm, " ");
setcolor(YELLOW);
circle(150, 200, 100);
circle(450, 200, 100);
setfillstyle(SOLID_FILL, YELLOW);
floodfill(150, 200, YELLOW);
floodfill(450, 200, YELLOW);
setcolor(DARKGRAY);
circle(100, 170, 20);
circle(100+100, 170, 20);
circle(100+300, 170, 20);
circle(100+400, 170, 20);
setfillstyle(SOLID_FILL, DARKGRAY);
floodfill(100, 170, DARKGRAY);
floodfill(100+100, 170, DARKGRAY);
floodfill(100+300, 170, DARKGRAY);
floodfill(100+400, 170, DARKGRAY);
arc(150, 200+25, 180, 360, 40);
arc(450, 200+65, 0, 180, 40);
getch();
closegraph();
}
```

Output



Program - 24

// WAP to draw Traffic light

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm," ");
  int radius = 35;
  setbkcolor(WHITE);
  setfillstyle(SOLID-FILL, DARKGRAY);
  bar(250, 10,400,300);
  bar(300, 300, 350, 400);
  bar(270, 400,380,430);
  setcolor(RED);
  circle(325, 60, radius);
  setfillstyle(SOLID-FILL,RED);
  floodfill(325, 60, RED);
  setcolor(GREEN);
  circle(325,150, radius);
  setfillstyle(SOLID-FILL, GREEN);
  floodfill(325, 150, GREEN);
  setcolor(YELLOW);
  circle(325, 240, radius);
  setfillstyle(SOLID-FILL, YELLOW);
  floodfill(325, 240, YELLOW);
  getch();
  closegraph();
}
```

Output -

Max X 639, Max Y 479

Program- 25

```
// WAP to use getmaxx(), getmaxy()

#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
  int gd= DETECT, gm, maxx, maxy;
  initgraph(&gd,&gm," ");
  maxx = getmaxx();
  maxy= getmaxy();
  char text[50];
  sprintf(text, " Max X %d , Max Y %d ", maxx, maxy);
  outtextxy (10, 10, text);
  rectangle(10, 30, maxx-10, maxy-10);
  getch();
  closegraph();
}
```

Output-

CODERJI

Program - 26

// WAP to draw name using line function

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd=DETECT, gm;
 int y1=150, y2=200, y3=250;
 initgraph(&gd, &gm, " ");
 setbkcolor(BLUE);
 line(50, y1, 50, y3);
 line(50, y1, 100, y1);
 line(50, y3, 100, y3);
 line(150, y1, 200, y1);
 line(150, y1, 150, y3);
 line(150, y3, 200, y3);
 line(200, y1, 200, y3);
 line(250, y1, 250, y3);
 line(240, y1, 300, y1+20);
 line(300, y1+20, 300, y3-20);
 line(240, y3, 300, y3-20);
 line(350, y1, 350, y3);
 line(350, y2, 400, y2);
 line(350, y3, 400, y3);
 line(450, y1, 450, y3);
 line(450, y1, 500, y1);
 line(500, y1, 500, y2);
 line(450, y2, 500, y2);
 line(450, y2, 500, y3);
 line(550, y1, 600, y1);
```

```
line(575, y1, 575, y3);
line(575, y3, 550, y3);
line(550, y2, 550, y3);
line(615, y1, 615, y3);
getch();
closegraph();
}
```

Output-

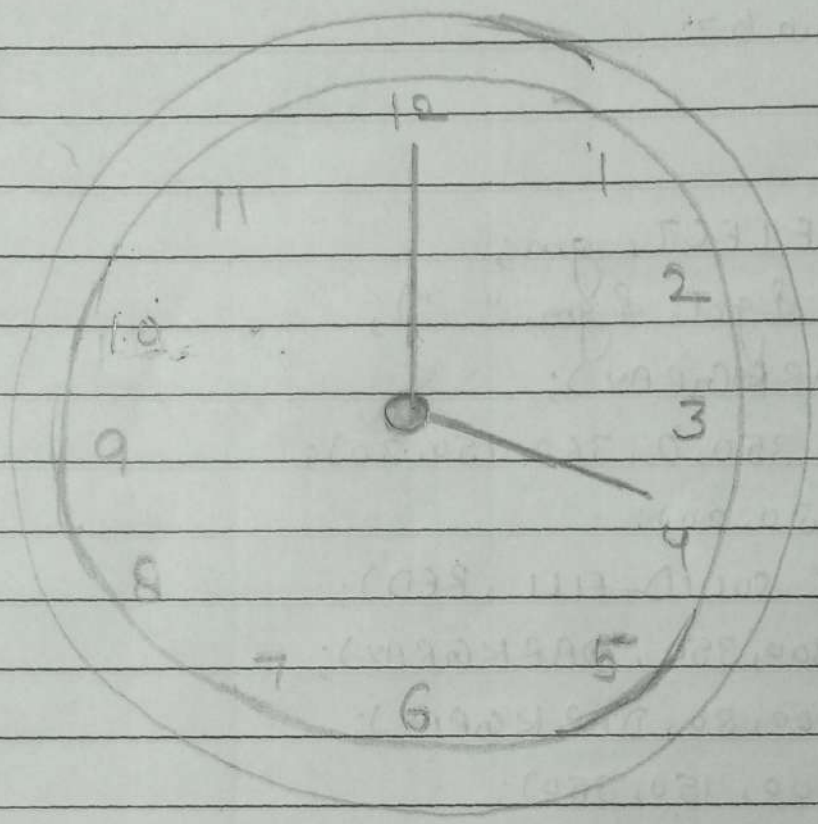
Program-27

// WAP to draw cap

input-

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  setcolor(DARKGRAY);
  ellipse(300, 350, 0, 360, 150, 30);
  circle(300, 80, 20);
  setfillstyle(SOLID-FILL, RED);
  floodfill(300, 350, DARKGRAY);
  floodfill(300, 80, DARKGRAY);
  line(300, 100, 150, 350);
  line(300, 100, 150+300, 350);
  setfillstyle(SOLID-FILL, YELLOW);
  floodfill(200, 300, DARKGRAY);
  getch();
  closegraph();
}
```

Output -



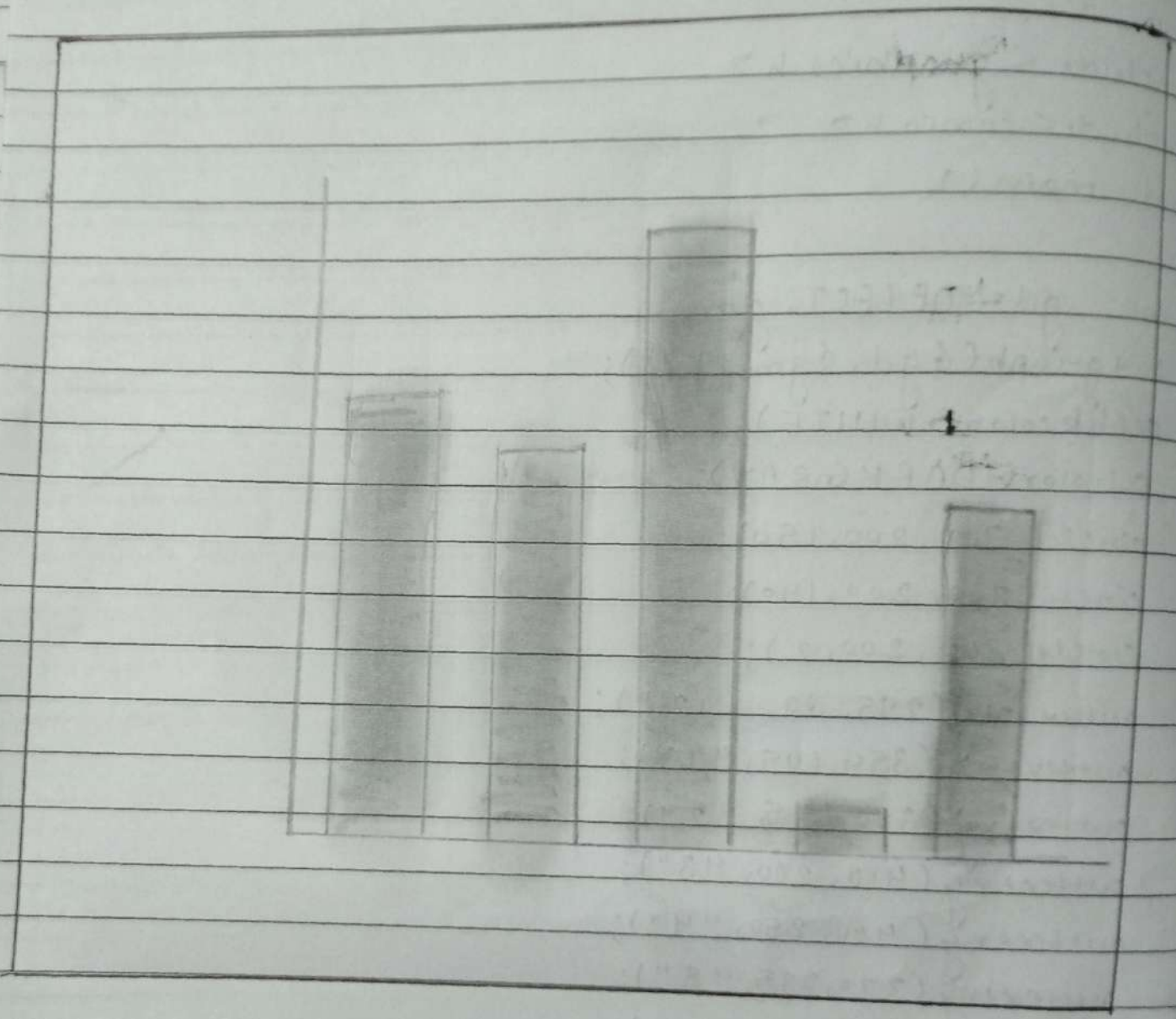
Program-28

// WAP to draw clock

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  setbkcolor(WHITE);
  setcolor(DARKGRAY);
  circle(300, 200, 150);
  circle(300, 200, 140);
  circle(300, 200, 2);
  outtextxy(295, 80, "12");
  outtextxy(359 105, "1");
  outtextxy(390, 145, "2");
  outtextxy(410, 200, "3");
  outtextxy(400, 250, "4");
  outtextxy(370, 295, "5");
  outtextxy(308, 320, "6");
  outtextxy(230, 298, "7");
  outtextxy(190, 250, "8");
  outtextxy(180, 200, "9");
  outtextxy(190, 140, "10");
  outtextxy(227, 98, "11");
  line(300, 100, 300, 200);
  line(300, 200, 390, 220);
  getch();
  closegraph();
}
```

output-



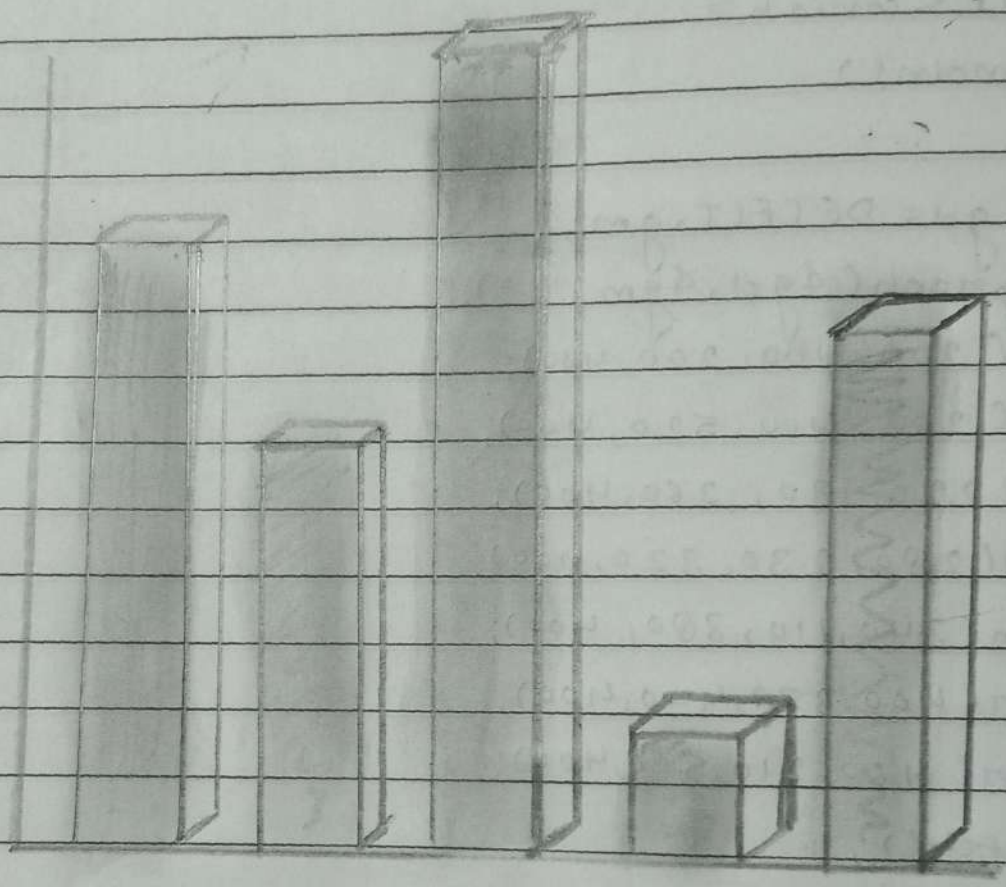
Program - 29

WAP to draw bargraph using bar function

Input-

```
#include <graphics.h>
#include <conio.h>
void main()
{
 int gd=DETECT,gm;
 initgraph(&gd,&gm,"");
 line(200,100,200,400);
 line(200,400,520,400);
 bar(220,190,260,400);
 bar(280,230,320,400);
 bar(340,110,380,400);
 bar(400,370,440,400);
 bar(460,210,500,400);
 getch();
closegraph();
}
```

output -



Program-30

// WAP to draw 3d bargraph using bar3d() function

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd=DETECT, gm;
 initgraph(&gd, &gm, " ");
 line(200, 100, 200, 400);
 line(200, 400, 520, 400);
 bar3d(220, 190, 260, 400, 10, 1);
 bar3d(280, 230, 320, 400, 10, 1);
 bar3d(340, 110, 380, 400, 10, 1);
 bar3d(400, 370, 440, 400, 10, 1);
 bar3d(460, 210, 500, 400, 10, 1);
 getch();
 closegraph();
}
```

output-

Date
Page No.

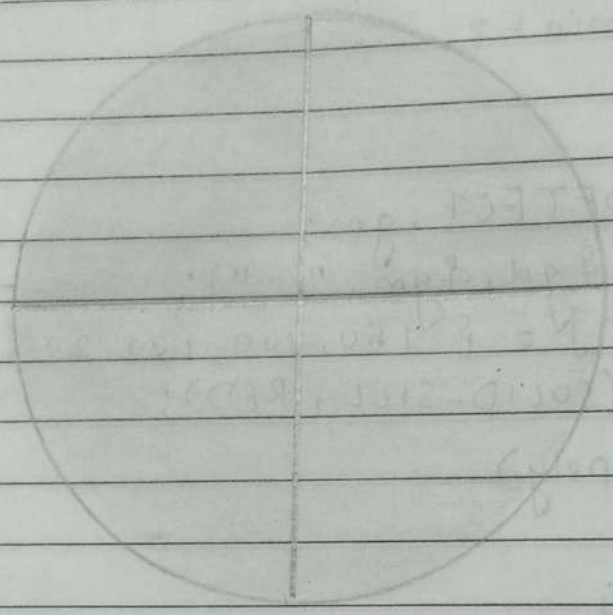
Program - 31

// WAP to use fillpoly()

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  int poly[] = {150, 100, 100, 200, 200, 150, 100};
  setfillstyle (SOLID-SILL, RED);
  fillpoly(3, poly)
  getch();
  closegraph();
}
```

Output-



Program - 32

// WAP to draw pie-chart.

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
    int gd = DETECT , gm;
    initgraph(&gd, &gm, " ");
    int xc = 250, yc = 200, radius = 200;
    setfillstyle(SOLID-FILL, RED);
    sector(xc, yc, 0, 90, radius, radius);
    setfillstyle(SOLID-FILL, GREEN);
    sector(xc, yc, 90, 180, radius, radius);
    setfillstyle(SOLID-FILL, BLUE);
    sector(xc, yc, 180, 270, radius, radius);
    setfillstyle(SOLID-FILL, YELLOW);
    sector(xc, yc, 270, 360, radius, radius);
    getch();
    closegraph();
}
```

Output

Program-33

// WAP to use pieslice()

Syntax - pieslice(x, y, S_A, E_A, radius);

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, " ");
    int xc=250, yc=200, radius=100;
    setfillstyle(SOLID_FILL, RED);
    pieslice(xc, yc, 0, 90, radius);
    getch();
    closegraph();
}
```

output -

Date / /
Page No.

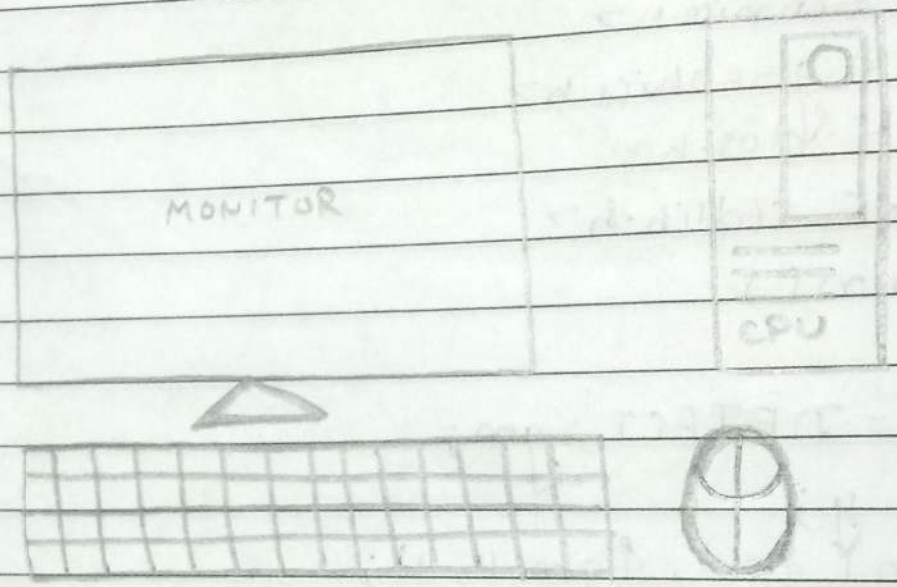
Program-34

// WAP to generate circle on screen randomly

input-

```
#include< stdio.h>
#include <conio.h>
#include< graphics.h>
#include < dos.h>
#include< stdlib.h>
void main()
{
 int gd=DETECT, gm;
 int x, y;
 initgraph(&gd, &gm," ");
 x=getmaxx();
  y=getmaxy();
 while(!kbhit())
 {
 setcolor(random(100));
 circle(random(x), random(y), random (100));
 delay(100);
 }
 getch();
  closegraph();
}
```

1.01 Output -



Program - 35

//WAP to draw computer

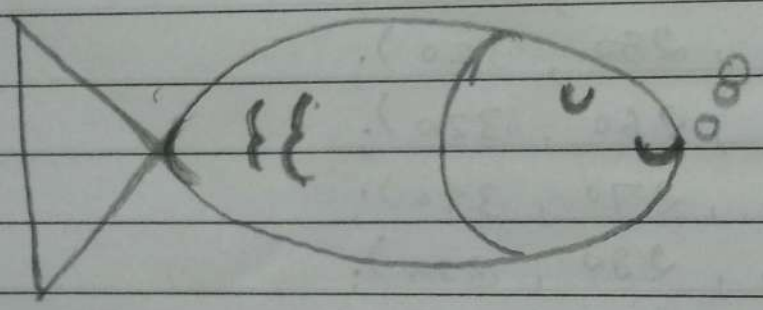
input -

```
#include< graphics.h>
#include < conio.h>
void main()
{
int gd= DETECT, gm;
initgraph(&gd, &gm," ");
rectangle(100, 100, 300, 250);
outtextxy(175, 175, " MONITOR");
line(200, 250, 180, 270);
line(200, 250, 220, 270);
line(180, 270, 220, 270);
rectangle(350,100,400, 250);
rectangle (370,110,390,170);
outtextxy(360,220,"CPU");
circle(380, 150, 5);
line(360, 190, 390, 190);
line(360, 200, 390, 200);
line(360, 210, 390, 210);
line( 100, 290, 320, 290);
rectangle(100, 280, 320, 320);
line(100,300, 320, 300);
line(100, 310, 320, 310);
line(120, 280, 120, 320);
line(130, 280, 130, 320);
line (140, 280, 140, 320);
line(150, 280, 150, 320);
line(160, 280, 160, 320);
line(170, 280, 170, 320);
```

mid

```
line(180, 280, 180, 320);
line(190, 280, 190, 320);
line(200, 280, 200, 320);
line(210, 280, 210, 320);
line(220, 280, 220, 320);
line(230, 280, 230, 320);
line(240, 280, 240, 320);
line(250, 280, 250, 320);
line(260, 280, 260, 320);
line(270, 280, 270, 320);
line(280, 280, 280, 320);
line(290, 280, 290, 320);
line(300, 280, 300, 320);
line(310, 280, 310, 320);
line(320, 280, 320, 320);
ellipse(350, 300, 0, 360, 15, 20);
line(350, 280, 350, 320);
arc(350, 290, 180, 0, 12);
getch();
closegraph();
}
```

utput -



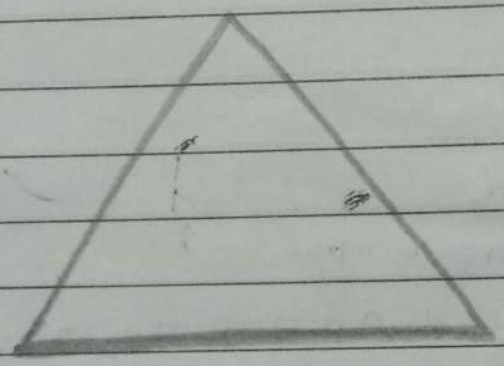
Program-36

// WAP to draw fish

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  int poly[] = {138, 168, 90, 110, 90, 230, 138, 168};
  ellipse(250, 200, 20, 160, 120, 90);
  ellipse(250, 138, 200, 340, 120, 90);
  arc(320, 152, 180, 360, 9);
  arc(353, 160, 230, 310, 15);
  arc(300, 168, 120, 240, 67);
  arc(215, 155, 120, 230, 15);
  arc(215, 180, 120, 230, 15);
  arc(190, 161, 120, 230, 8);
  arc(190, 174, 120, 230, 8);
  circle(360, 140, 5);
  circle(370; 125, 7);
  circle(360, 110, 8);
  drawpoly(4, poly);
  getch();
  closegraph();
}
```

Output -



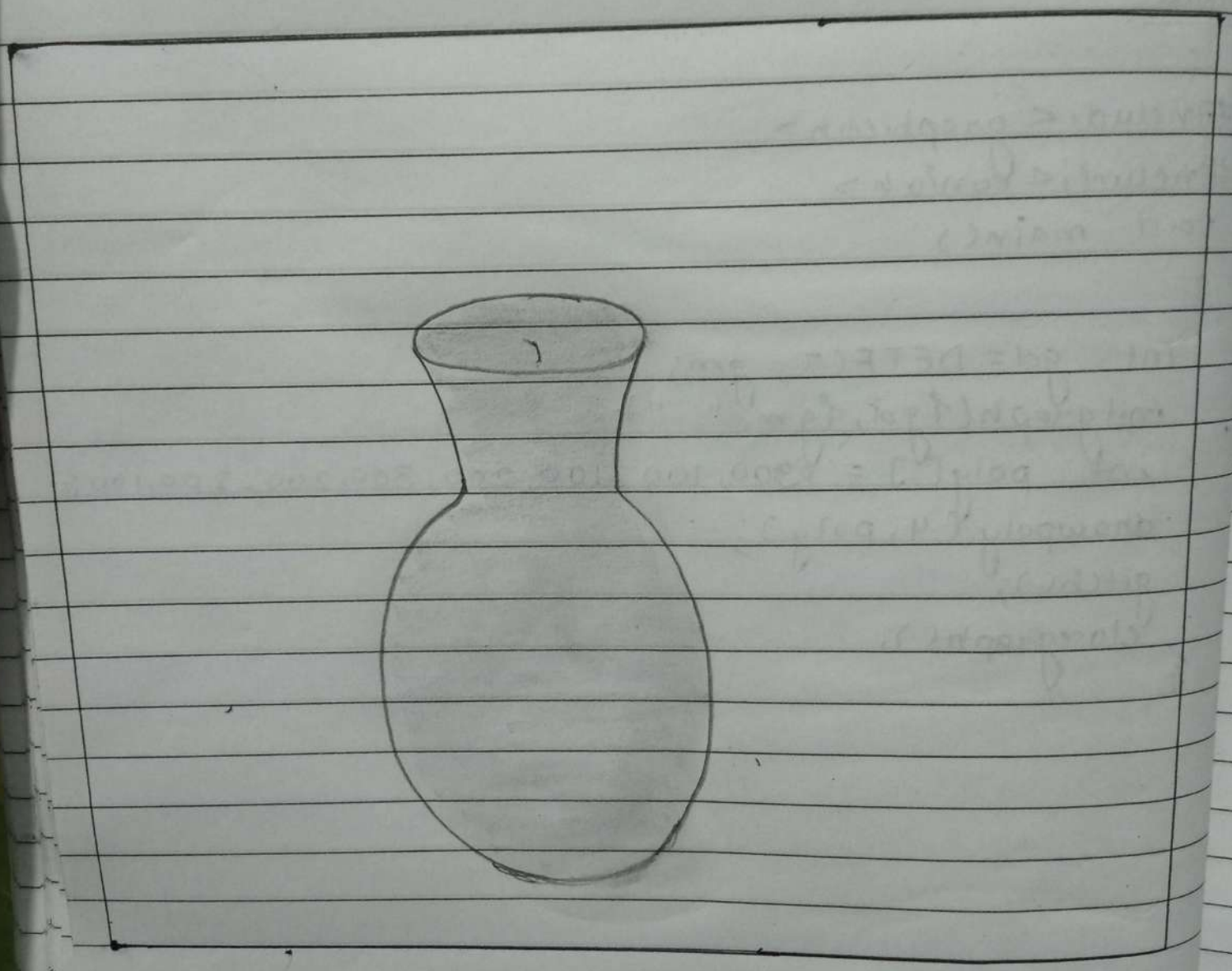
Program - 37

// WAP to draw triangle using drawpoly() function

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  int poly[] = {200, 100, 100, 200, 300, 200, 200, 100};
  drawpoly(4, poly);
  getch();
  closegraph();
}
```

output -



Program - 38

// WAP to draw pot.

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd=DETECT,gm;
  initgraph(&gd,&gm," ");
  setcolor(YELLOW);
  ellipse(300,150,0,360,60,20);
  arc(122,245,359,35,150);
  arc(450,230,140,190,120);
  ellipse(300,350,0,360,80,110);
  setfillstyle(SOLID_FILL,YELLOW);
  floodfill(300,200,YELLOW);
  floodfill(300,350,YELLOW);
  getch();
  closegraph();
}
```

output-

Program - 39

//WAP to draw flower pot

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd=DETECT, gm;
 initgraph(&gd, &gm, " ");
 int i=100, j=50;
  //Flower1
  setfillstyle (SOLID-FILL, 5);
  pieslice (250,100,0,30,40);
  pieslice (250,100,50,80,40);
  pieslice (250,100,100,130,40);
  pieslice (250,100,150,180,40);
  pieslice (250,100,210,240,40);
  pieslice (250,100,260,290,40);
  pieslice (250,100,310,340,40);
  setcolor (GREEN);
  line (250,100,350,350);
  setcolor (WHITE);
  //Flower 2
  setfillstyle(SOLID-FILL,14);
  pieslice (250+i,100+j, 0,30,40);
  pieslice (250+i,100+j,50,80,40);
  pieslice (250+i,100+j,100,130,40);
  pieslice (250+i,100+j,150,180,40);
  pieslice (250+i,100+j,210,240,40);
  pieslice (250+i,100+j,260,290,40);
  pieslice (250+i,100+j,310,340,40);
```

```
setcolor( GREEN);
line( 250+i , 100+j , 350, 350);
 setcolor( WHITE);
 i= 200, j=0;
 //Flower 3
 setfillstyle( SOLID-FILL, 13);
 pieslice (250+ i, 100+j, 0, 30, 40);
 pieslice ( 250+i, 100+j, 50, 80, 40);
 pieslice( 250 +i, 100+ j, 100, 130, 40);
 pieslice ( 250 + i, 100 + j, 150, 180, 40);
 pieslice (250+i, 100+j, 210, 240, 40);
 pieslice ( 250+i, 100+j, 260, 290, 40);
 pieslice (250+i, 100+j, 310, 340, 40);
 setcolor( GREEN);
 line( 250+ i, 100+j ,350, 350);
 setcolor( WHITE);
 //POT
 ellipse( 350, 350, 20, 160, 30, 10);
 ellipse( 350, 344, 200, 350, 30, 10);
 setfillstyle (SOLID-FILL, BROWN);
 floodfill( 350, 345, WHITE);
 ellipse (380, 375, 120, 250, 10, 30);
 ellipse( 317, 374, 290, 60, 10, 30);
 ellipse( 348, 430, 130, 55, 40, 40);
 setfillstyle ( LINE-FILL, CYAN);
 floodfill (348, 430, WHITE);
  setcolor (11);
  settextstyle (8, 0, 3);
  outtextxy( 400, 400, "FLOWER POT");
 getch();
  closegraph();
}
```

output -

Tue Mar 25 11:06:43

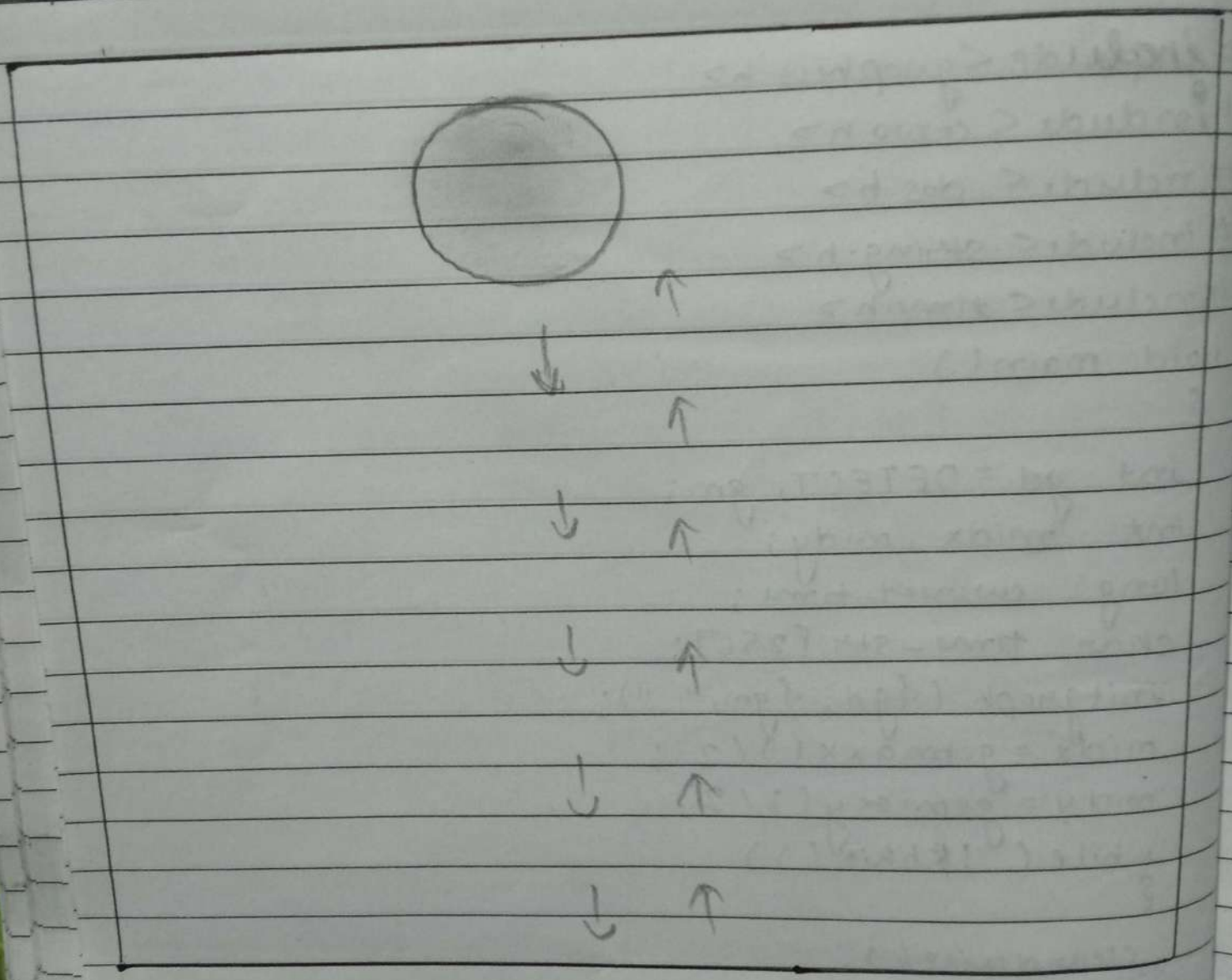
Program-40

//WAP to draw digital clock.

input -

```
#include <graphics.h>
#include < conio.h>
#include < dos.h>
#include< string.h>
#include< time.h>
void main()
{
  int gd =DETECT, gm;
  int midx , midy;
  long current_time;
  char time_str [256];
  initgraph (&gd, &gm," ");
  midx = getmaxx()/2 ;
  midy = getmaxy()/2;
  while( !kbhit())
  {
   cleardevice();
   current_time = time(NULL);
   strcpy(time_str, ctime(&current_time);
   setcolor(2);
   settextstyle( DEFAULT_FONT,HORIZ_DIR, 4);
   outtextxy (midx-300, midy, time_str);
   delay(1000);
  }
  getch();
  closegraph();
}
```

Output-



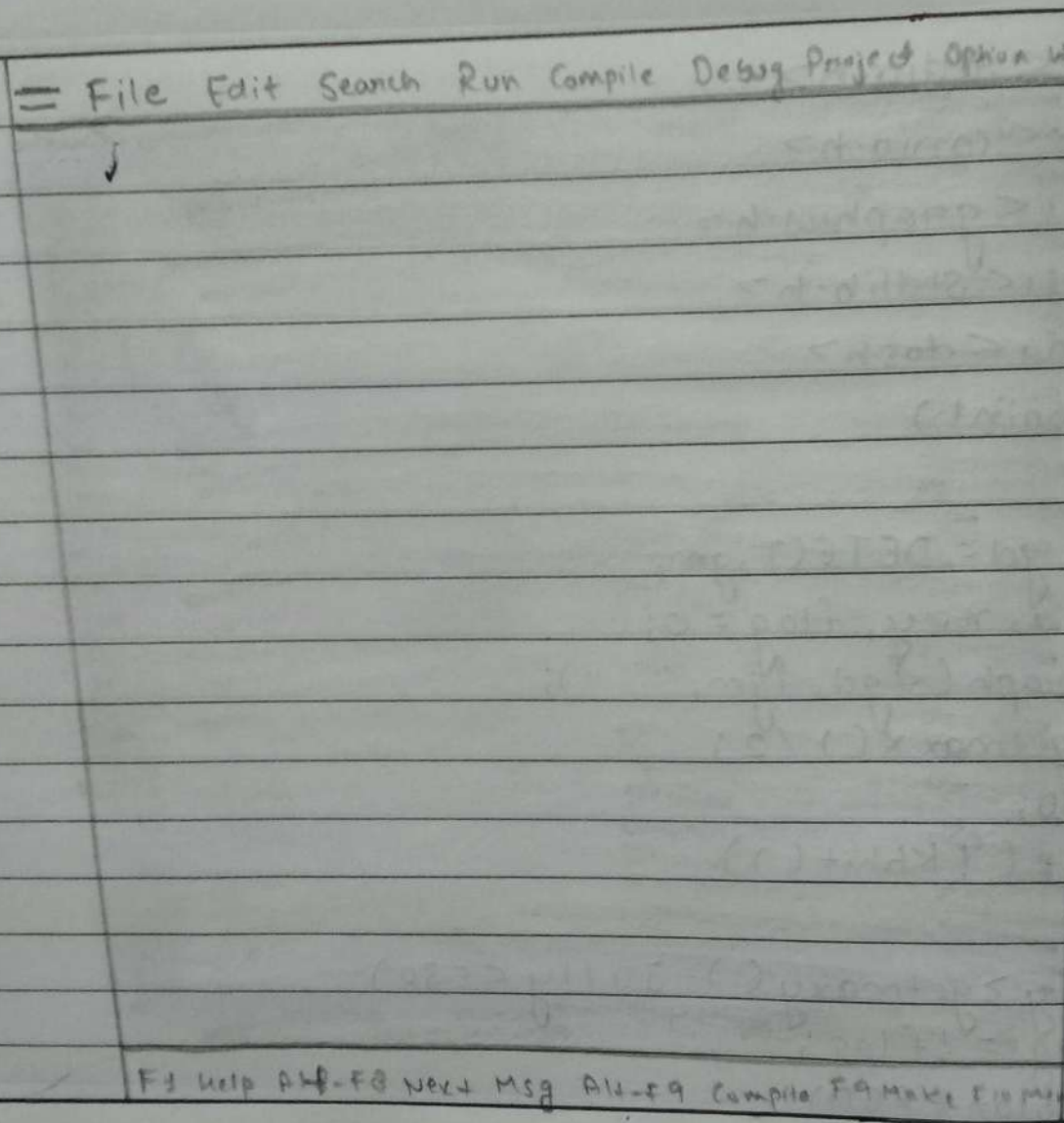
Program-41

// WAP to draw bouncing ball.

input-

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<stdlib.h>
#include<dos.h>
void main()
{
int gd=DETECT,gm;
int i,x,y,flag=0;
initgraph(&gd,&gm," ");
x=getmaxx()/2;
y=30;
while(!Kbhit())
{
if(y>getmaxy()-30||y<=30)
flag=!flag;
setcolor(RED);
setfillstyle(SOLID-FILL,RED);
circle(x,y,30);
floodfill(x,y,RED);
delay(50);
cleardevice();
if(flag)
{
y=y+15;}
else { y=y-15;} }
getch();
closegraph();
}
```

Output -


File Edit Search Run Compile Debug Project Options Win
F1 Help Alt-F8 Next Msg Alt-F9 Compile F9 Make F10 Menu

Program-42

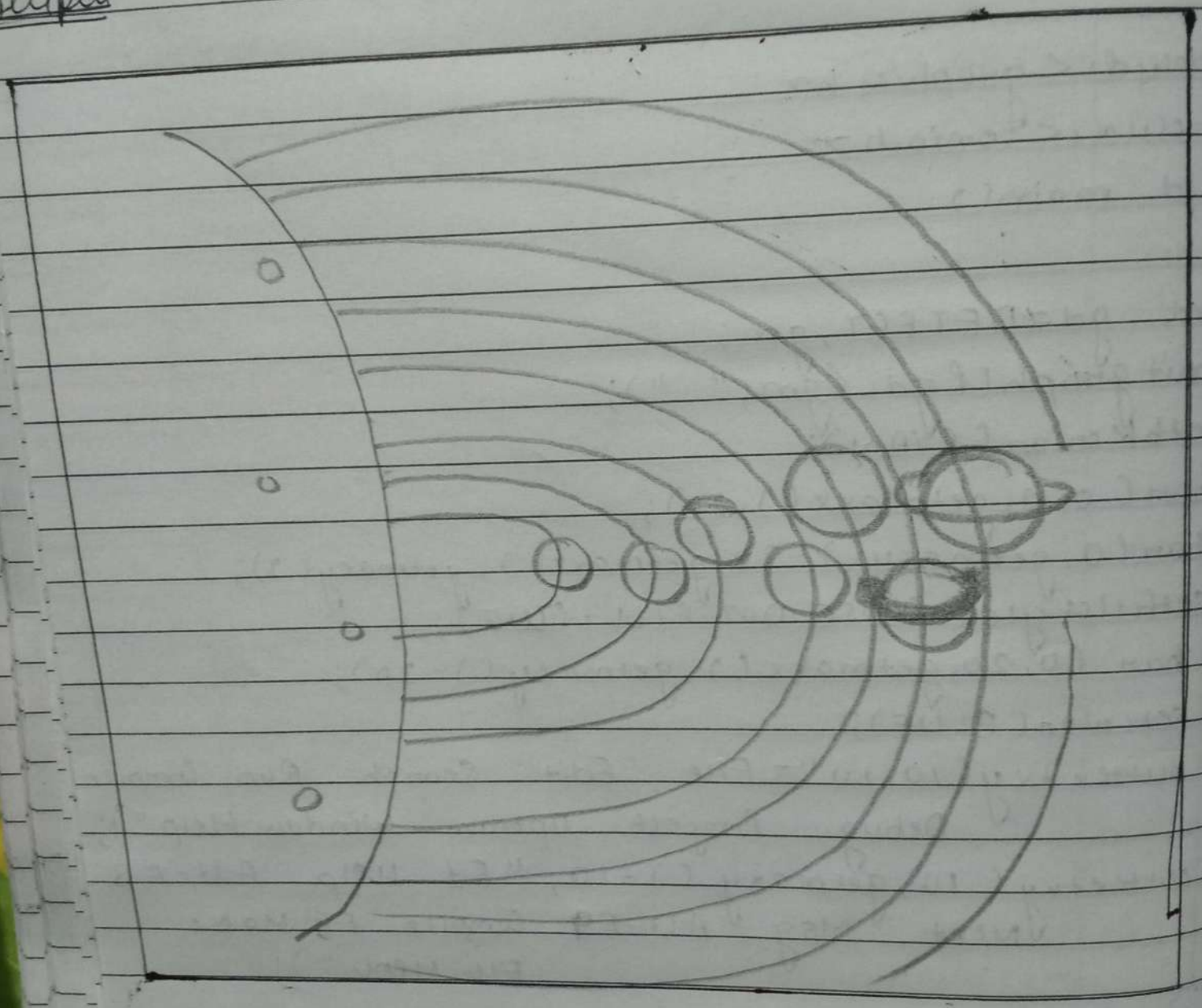
// WAP to draw TURBOC3 SCREEN.

input-

```
#include<graphics.h>
#include<conio.h>
 void main()
 {
  int gd=DETECT,gm;
  initgraph(&gd,&gm," ");
  setbkcolor(CYAN);
   bar(0,0,getmaxx(),20);
   bar(0,getmaxy()-20,getmaxx(),getmaxy());
   setfillstyle(CLOSE_DOT_FILL,BLUE);
    bar(0,20,getmaxx(),getmaxy()-20);
    setcolor(BLUE);
    outtextxy(10,10"= File  Edit  Search  Run  Compile
              Debug  Project  Option  Window  Help");
    outtextxy(10,getmaxy()-10,"F1 Help  Alt-F8
              Next Msg  Alt-F9 Compile  F9 Make
                                F10 Menu");

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

Output -



Program-43

// WAP to draw Solar system.

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm," ");
  arc(0, getmaxy()/2, 270, 90, 200);
  circle(10, 150, 10);
  circle(100, 100, 10);
  circle(150, 200, 10);
  circle(150, 300, 10);
  circle(50, 250, 10);
  circle(50, 360, 10);
  ellipse(200, getmaxy()/2, 266, 92, 100, 50);
  circle(300, 240, 10);
  ellipse(200, getmaxy()/2, 266, 94, 150, 70);
  circle(350, 240, 13);
  ellipse(350, 240,
  ellipse(200, getmaxy()/2, 264, 96, 200, 90);
  circle(400, 240, 15);
  ellipse(200, getmaxy()/2, 262, 97, 250, 110);
  circle(450, 240, 18);
  ellipse(200, getmaxy()/2, 261, 99, 300, 130);
  circle(495, 220, 25);
  ellipse(200, getmaxy()/2, 260, 101, 350, 150);
  circle(550, 260, 20)
  ellipse(550, 260, 188, 2, 25, 5);
  ellipse(550, 260, 188, 2, 25, 7);
```

```
ellipse(200, getmaxy()/2, 257, 103, 400, 170);
circle(600, 220, 10);
ellipse(600, 220, 145, 8, 25, 10);

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

Output -

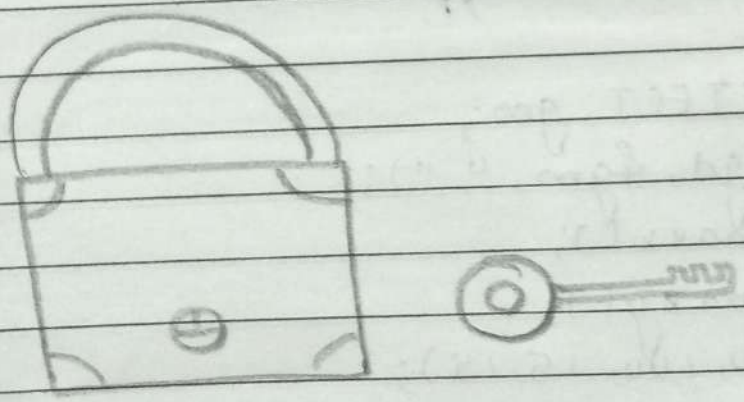
Program-44

// WAP to draw Night sky

input-

```
#include <graphics.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
void main()
{
  int gd= DETECT, gm;
  initgraph(&gd, &gm, " ");
  int x=getmaxx();
  int y=getmaxy();
  fillellipse(50, 100, 15,15);
  for(int i=0; i<=200, i++)
  {
    putpixel(random(x),random(y), WHITE);
    delay (50);
  }
  getch();
  closegraph();
}
```

Output -



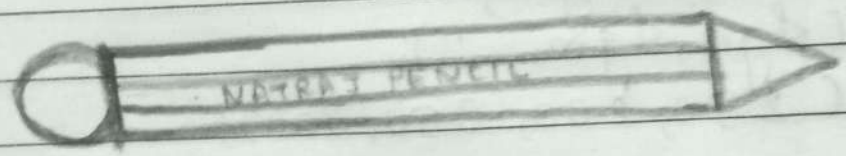
Program-45

// WAP to draw lock and key

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd,&gm," ");
  rectangle (250, 200, 350, 300);
  ellipse( 300, 200, 0, 180, 50, 80);
  ellipse( 300, 200, 0, 180, 40, 70);
  arc ( 298, 240, 300, 250, 5);
  line ( 296, 245, 296, 255);
  line ( 296+3, 245, 296+3, 255);
  line ( 296, 255, 299, 255);
  arc(400, 300, 310, 235, 15);
  circle(400, 300, 10);
  line(392, 312, 392, 380);
  line(392+15, 312, 392+15, 356);
  line( 392, 380, 398+15+5, 380);
  line(398+20, 380, 398+20, 375);
  line(398+20, 375, 405, 375);
  line(405, 375, 405, 370);
  line( 405, 370, 412, 370);
  line( 412, 370, 412, 366);
  line( 412, 366, 405, 366);
  line(405, 366, 405, 360);
  line( 405, 360, 418, 360);
  line( 418, 360, 418, 360);
  line( 418, 356, 407, 356);
getch();
closegraph();
}
```

Output -



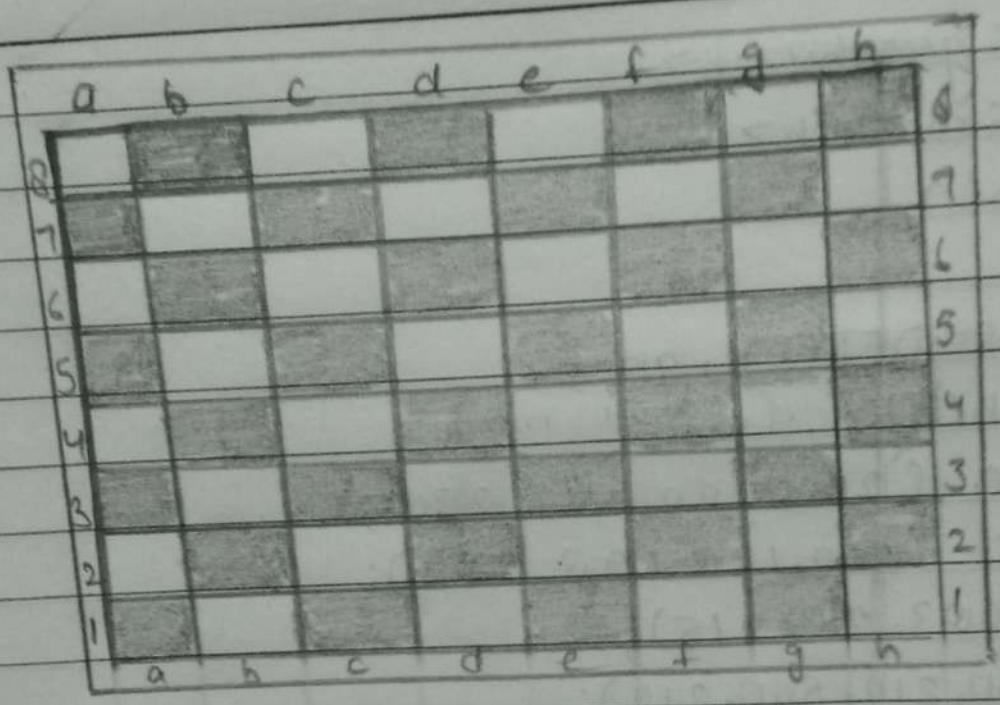
Program - 46

// WAP to draw pencil

input -

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  rectangle (200, 200, 500, 230);
  rectangle (200, 190, 190, 232);
  circle (193, 214, 15);
  line (200, 210, 500, 210);
  line (200, 220, 500, 220);
  outtextxy (300, 213, "NATRAJ PENCIL");
  line (500, 200, 540, 215);
  line (500, 230, 540, 215);
  getch();
  closegraph();
}
```

Output-



CHESS BOARD

Program-47

// WAP to draw chess board

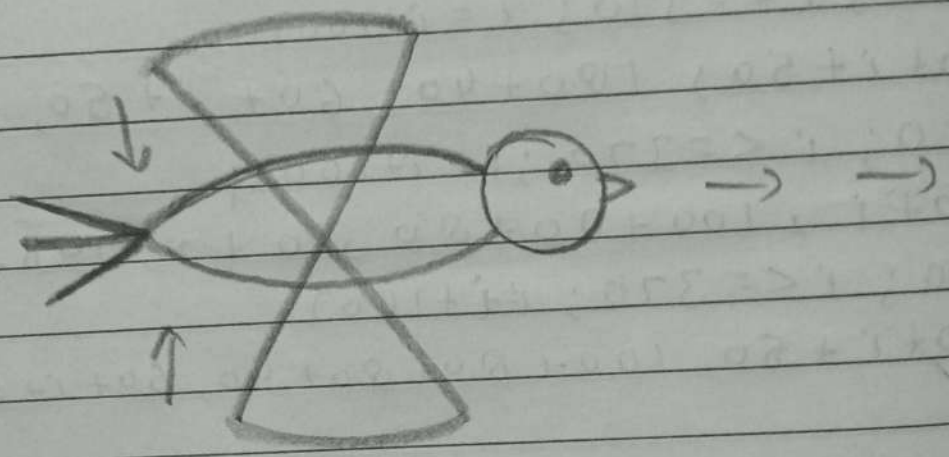
input-

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  rectangle(0, 10, 425, 350);
  rectangle(0, 20, 410, 340);
  outtextxy(15,12," a  b  c  d  e  f  g  h");
  outtextxy(15,342," a  b  c  d  e  f  g  h");
  outtextxy(2, 30," 8                    8");
  outtextxy(2,78," 7                     7");
  outtextxy(2,118," 6                    6");
  outtextxy(2, 118+40," 5                5");
  outtextxy(2,18+80," 4                  4");
  outtextxy(2,118+120," 3                3");
  outtextxy(2,118+160," 2                2");
  outtextxy(2,118+200," 1                1");
  for(int i=0; i<=370; i=i+100)
  bar(10+i, 20, 60+i, 60);
  for(i=0; i<=370; i=i+100)
  bar(10+50+i, 20+40, 60+i++50, 60+40);
  for(i=0; i<=370; i=i+100)
```

```
bar(10+i ,100, 60+i ,140);
for( i=0; i<=370; i=i+100)
bar( 10+i+50, 100+40, 60+i+50, 140+40);
for(i=0; i<=370; i=i+100)
bar (10+i , 180, 60+i, 220);
for( i=0; i<=370; i=i+100):
bar(10+i+50, 180+40, 60+i+50, 220+40);
for(i=0; i<=370 ;i=i+100){
bar(10+i , 100+80+80 ,60+i, 140+80+80);
for(i=0; i<=370; i=i+100)
bar(10+i+50, 100+80+80+40, 60+i+50, 140+80
                                    +80+40);

setcolor (11);
setfontstyle (0,0,3);
outtextxy (100, 380, " CHESS BOARD");
getch();
closegraph();
}
```

output-



Program-48

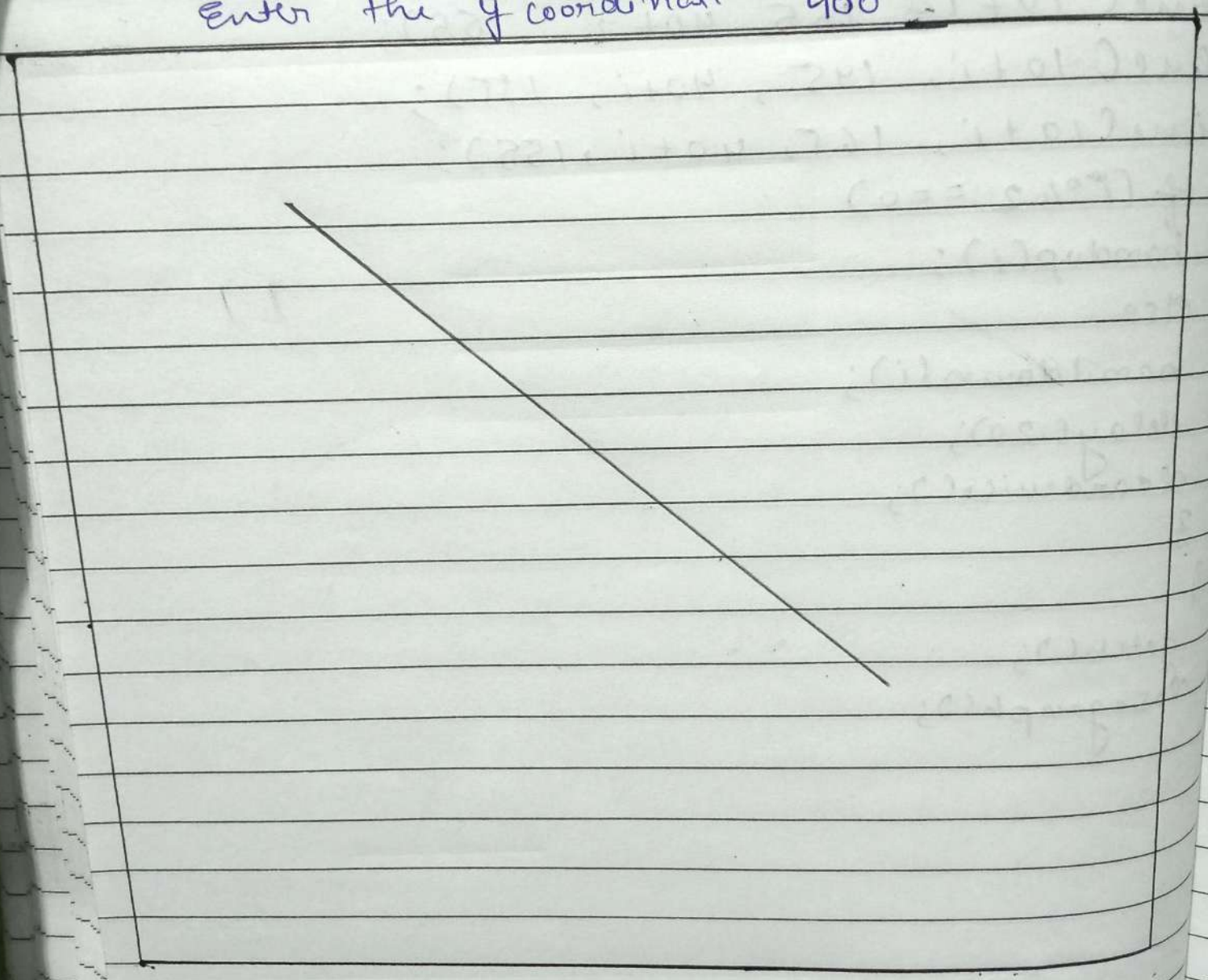
// WAP to draw Flying bird

input-

```
#include< graphics.h>
#include< conio.h >
#include< stdlib.h>
#include < dos.h >
 void  handdown (int i)
  {
   line( 85+i, 155, 45 +i, 185);
   line( 85+i, 155, 115+ i, 195);
   line( 90+ i, 130, 228, 290, 70);
  }
  void  handup (int i)
  {
   line ( 85+ i , 155 , 125+ i , 115);
   line ( 85+i, 155, 55+i , 118);
   arc ( 90+i, 177, 60, 122, 70);
  }
  void  main ( )
  {
    int  gd = DETECT, gm;
    initgraph ( &gd, &gm, " ");
    int  i=0 ;
    while( !kbhit())
    {
      for (i=0 ; i < 400; i++)
      {
       circle ( 150+i , 150, 20);
       arc(90+i , 190, 50, 145, 60);
       arc( 87+i , 117, 220, 320, 60);
```

```
line(170+i, 147, 180+i, 153);
line(180+i, 153, 170+i, 156);
circle(162+i, 150, 2);
line(10+i, 155, 40+i, 155);
line(10+i, 145, 40+i, 155);
line(10+i, 165, 40+i, 155);
 if (i%2 ==0)
 handup(i);
 else
 handdown(i);
 delay(20);
 cleardevice();
 }
}
 getch();
 closegraph();
}
```

Output- Enter the x coordinate = 100
Enter the y coordinate = 400



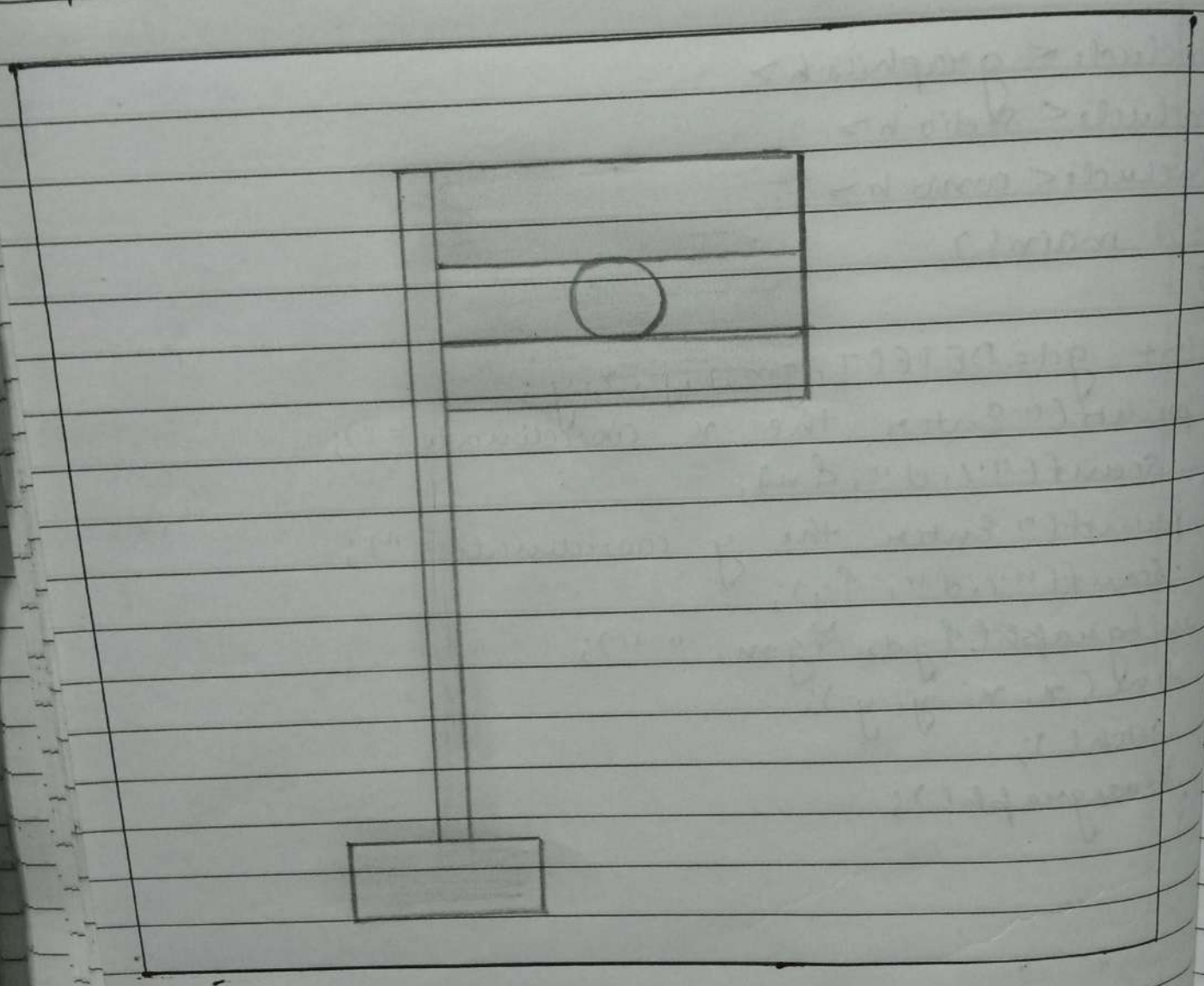
Program- 49

// WAP to draw a line dynamically.

input-

```
#include< graphics.h>
#include< stdio.h>
#include< conio.h>
void main()
{
  int gd=DETECT, gm, i, j, x, y;
  printf(" Enter the x coordinate =");
   scanf("%d", &x);
  printf(" Enter the y coordinate =");
   scanf("%d", &y);
  initgraph(&gd, &gm, " ");
   line(x, x, y, y);
   getch();
   closegraph();
}
```

output -



Program-50

WAP to draw Indian flag.

Input-

```
#include< graphics.h>
#include< conio.h>
#include< stdio.h>
#include< stdlib.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  setfillstyle (1, LIGHTRED);
  bar (200, 50, 450, 100);
  setfillstyle( 1, WHITE);
  bar( 200, 100, 450, 150);
  setfillstyle (1, GREEN);
  bar( 200, 150, 450, 200);
  setcolor(BLUE);
  circle (320, 125, 25);
  setfillstyle( 1, BLUE);
  fillellipse (320, 125, 15, 15);
  setfillstyle (1, BROWN);
  bar( 190, 40, 200, 430);
  setfillstyle (1, BROWN);
  getch();
  closegraph();
}
```

output-

Program - 51

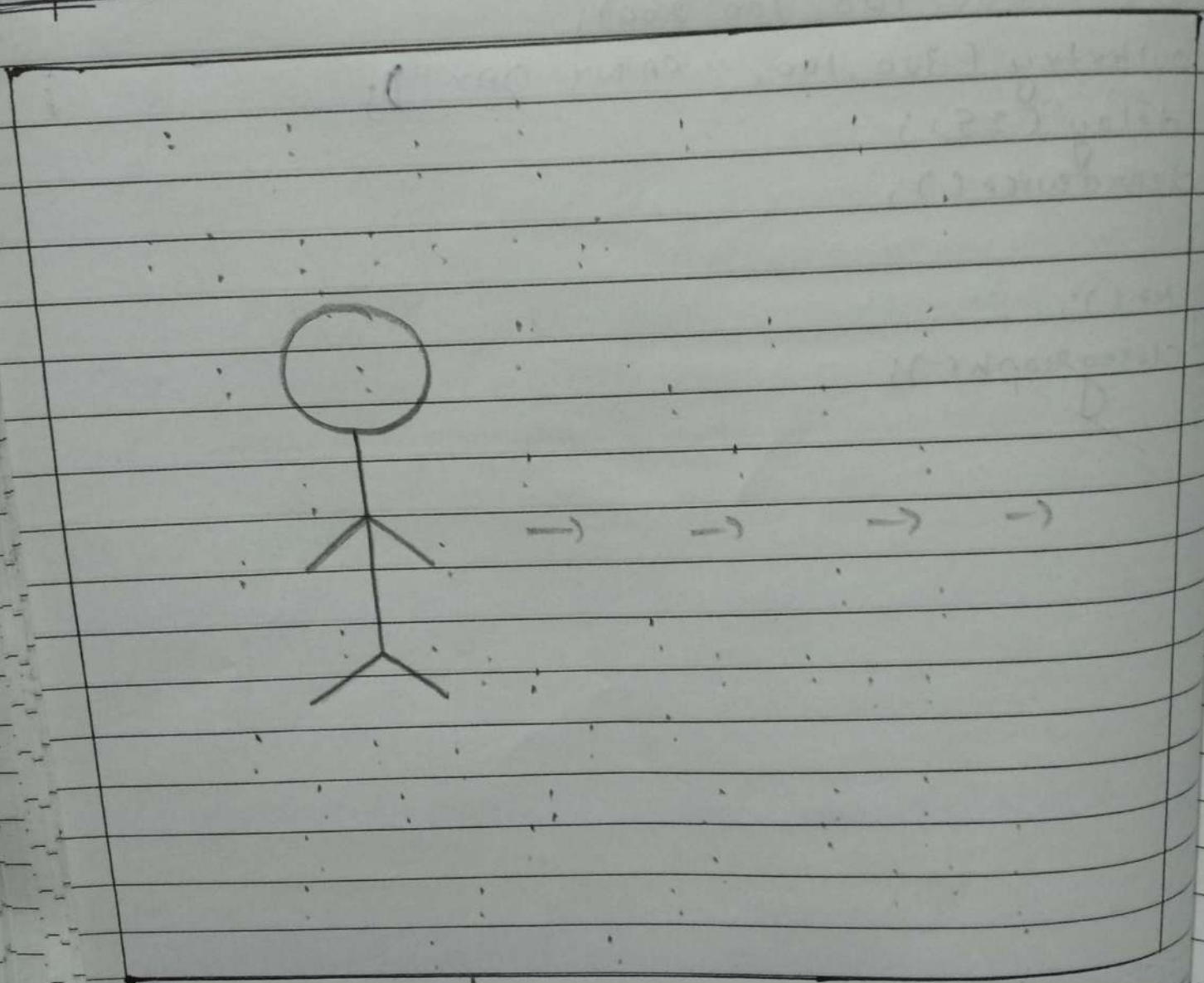
// WAP to draw rainy day umbrella

input -

```
#include < stdio.h >
#include< conio.h>
#include< stdlib.h>
#include< graphics.h>
#incude < dos.h>
void main()
{
  int gd= DETECT, gm, i, j, x, y;
  initgraph(&gd, &gm, " ");
  for(i=0; i< 100; i++)
  {
    x = getmaxx();
    y = getmaxy();
    for(j=0 ; j< =100; j++)
    {
     outtextxy(random(x), random(y), ":");
    }
    setbkcolor( WHITE);
    setcolor( DARKGRAY);
    circle( 200, 95, 5);
    ellipse( 200,200, 0, 180, 100,100);
    line( 200, 200, 300, 200);
    line( 200, 200, 200, 350);
    line( 190, 200, 190, 350);
    ellipse( 180, 350, 180, 0, 10, 10);
    ellipse( 185, 351, 180, 0, 15, 15);
    line( 200, 100, 300, 200);
    line( 200, 100, 250, 200);
```

```
line(200, 100, 210, 200);
line(200, 100, 170, 200);
line(200, 100, 130, 200);
line(200, 100, 100, 200);
outtextxy(300, 100, "RAINY DAY");
delay(25);
cleardevice();
}
getch();
closegraph();
}
```

output-



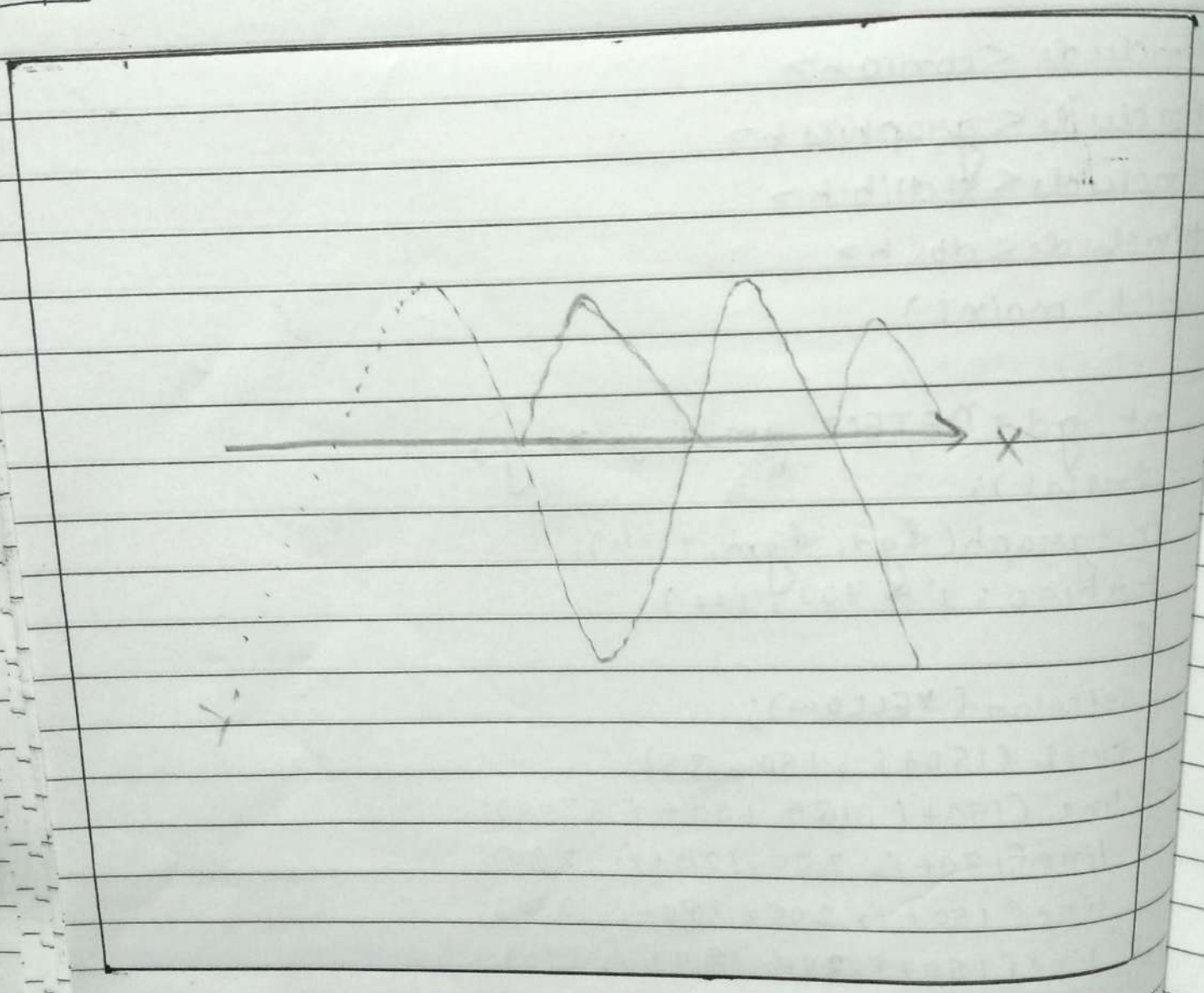
// Program - 52

// WAP to draw man object

input-

```
#include <conio.h>
#include <graphics.h>
#include <stdlib.h>
#include <dos.h>
void main()
{
int gd = DETECT, gm, i, j, x, y;
clrscr();
initgraph(&gd, &gm, " ");
for(i=0; i<400; i++)
{
setcolor(YELLOW);
circle(150+i, 150, 35);
line(150+i, 185, 150+i, 300);
line(150+i, 200, 120+i, 230);
line(150+i, 200, 180+i, 230);
line(150+i, 300, 120+i, 330);
line(150+i, 300, 180+i, 330);
outtextxy(230+i, 350, "Hi Everyone");
x = getmaxx();
y = getmaxy();
for(j=0; j<=100; j++)
outtextxy(random(x), random(y), ":");
delay(20);
cleardevice();
}
getch();
closegraph();
}
```

Output-



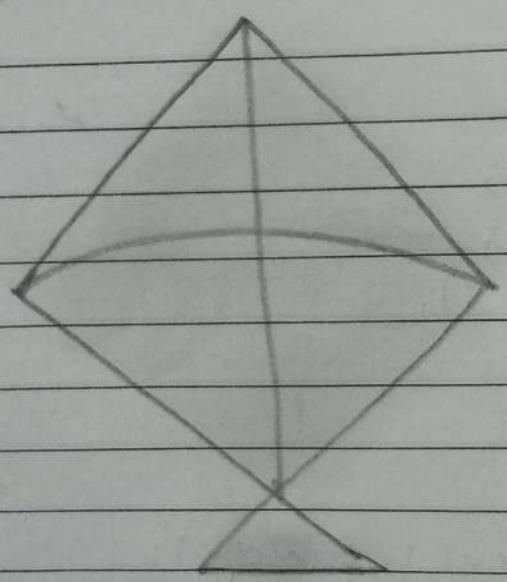
Program-53

// WAP to draw cos waves

input-

```
#include<graphics.h>
#include<conio.h>
#include<math.h>
void main()
{
  int gd=DETECT, gm;
  int x, y, midx, midy;
  initgraph(&gd, &gm, " ");
  midx = getmaxx()/2;
  midy = getmaxy()/2;
  setcolor(WHITE);
  line(0, midy, getmaxx(), midy);
  line(midx, 0, midx, getmaxy());
  setcolor(YELLOW);
  for(x=0; x<getmaxx(); x++)
  {
    y= midy - (50 * cos(x * 3.14/100));
    putpixel(x, y, YELLOW);
  }
  getch();
  closegraph();
}
```

output-



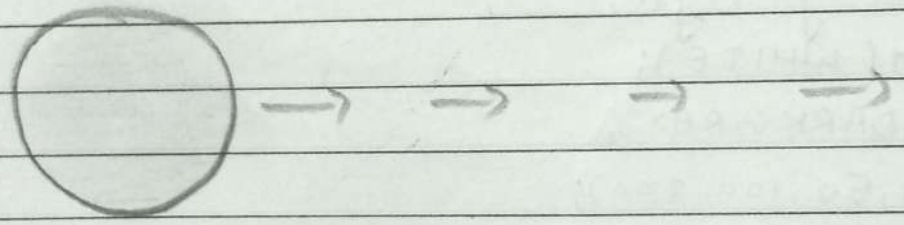
Program-54

// WAP to draw Kite

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd=DETECT,gm;
  initgraph(&gd,&gm," ");
  setbkcolor(WHITE);
  setcolor(DARKGRAY);
  line(250,50,100,200);
  line(100,200,250,330);
  line(250,50,400,200);
  setfillstyle(SOLID-FILL,CYAN);
  floodfill(250,300,DARKGRAY);
  line(250,330,210,270);
  line(250,330,290,370);
  line(210,370,290,370);
  setfillstyle(SOLID-FILL,CYAN);
  floodfill(240,350,DARKGRAY);
  arc(250,333,41,130,200);
  line(250,50,250,330);
  getch();
  closegraph();
}
```

Output-



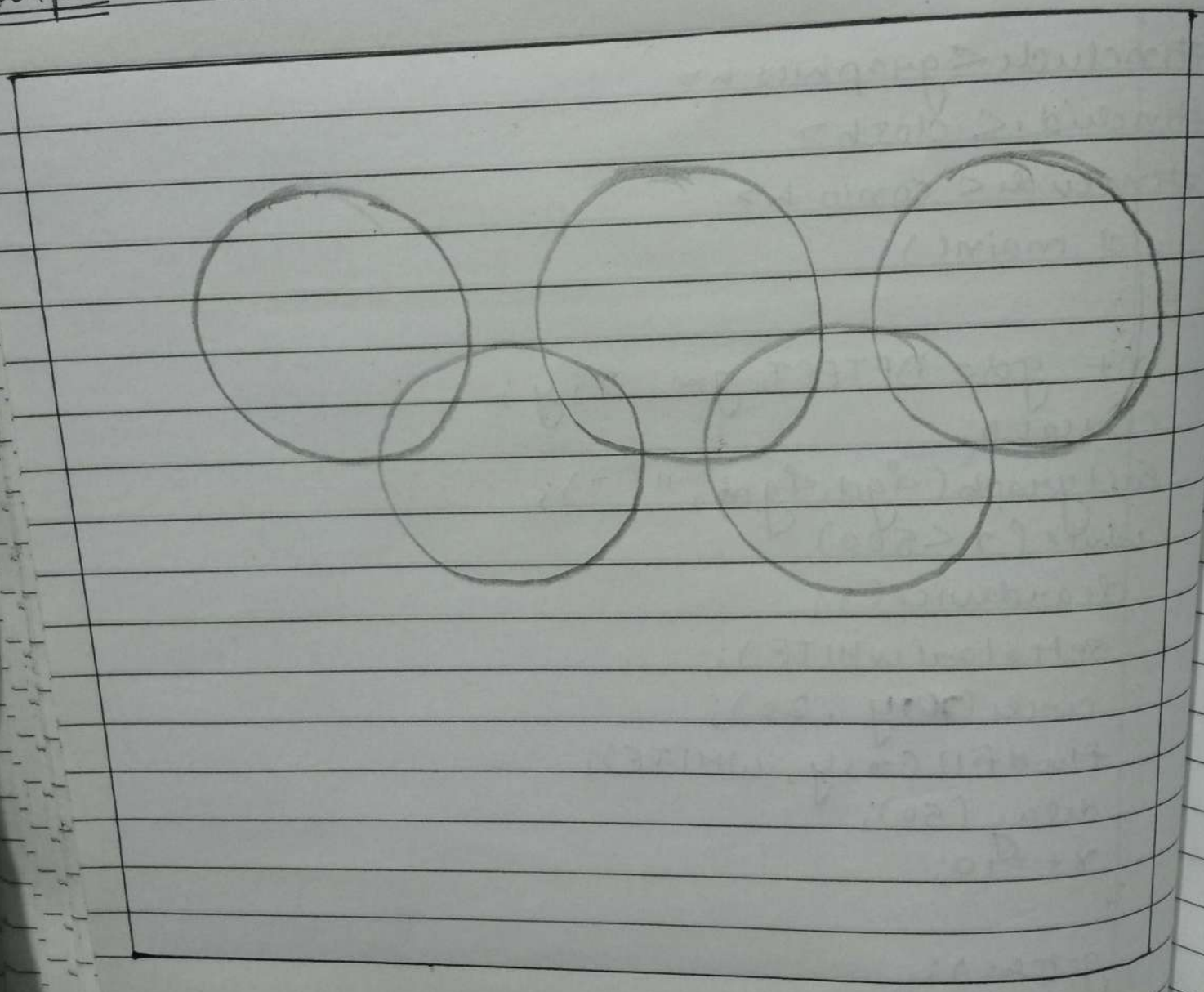
Program - 55

// WAP to move any object

input -

```
#include<graphics.h>
#include<dos.h>
#include<conio.h>
void main()
{
  int gd= DETECT, gm, x, y;
  clrscr();
  initgraph(&gd,&gm," ");
  while(x<500)
   cleardevice();
    setcolor(WHITE);
    circle(x,y,20);
    floodfill(x,y,WHITE);
    delay(50);
    x+=10;
  }
   getch();
   closegraph();
}
```

output



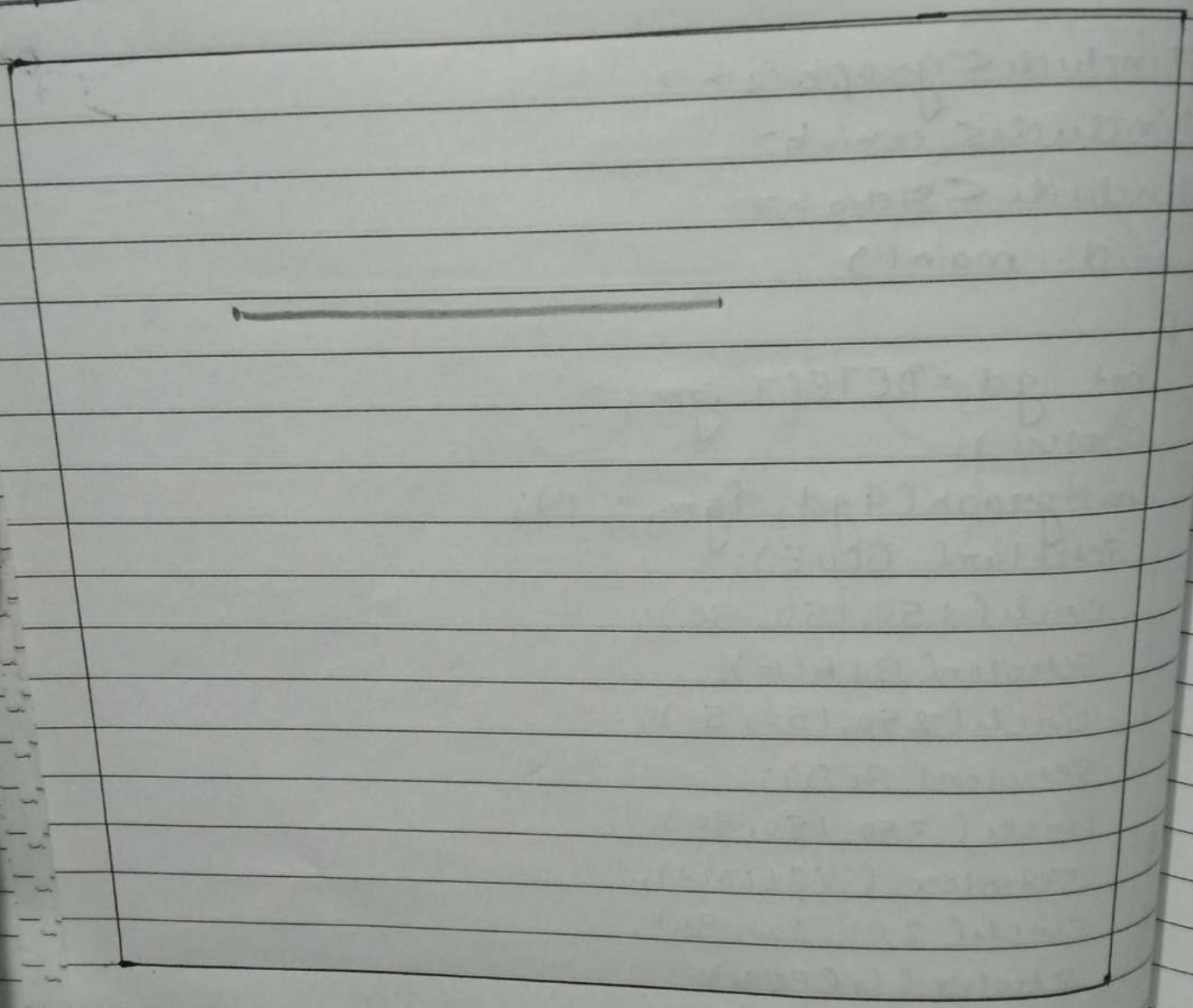
Program - 56

//WAP to draw olympic ring

input -

```
#include< graphics.h >
#include< conio.h>
#include < stdio.h>
void main()
{
  int gd =DETECT ,gm;
  clrscr();
  initgraph(&gd, &gm, " ");
  setcolor( BLUE);
  circle(150,150, 50);
  setcolor( BLACK);
  circle(250,150, 50);
  setcolor( RED);
  circle( 350,150, 50);
  setcolor (YELLOW);
  circle( 200, 200, 50);
  setcolor ( GREEN);
  circle( 300, 200, 50);
  getch();
  closegraph();
}
```

output-



Kanishk

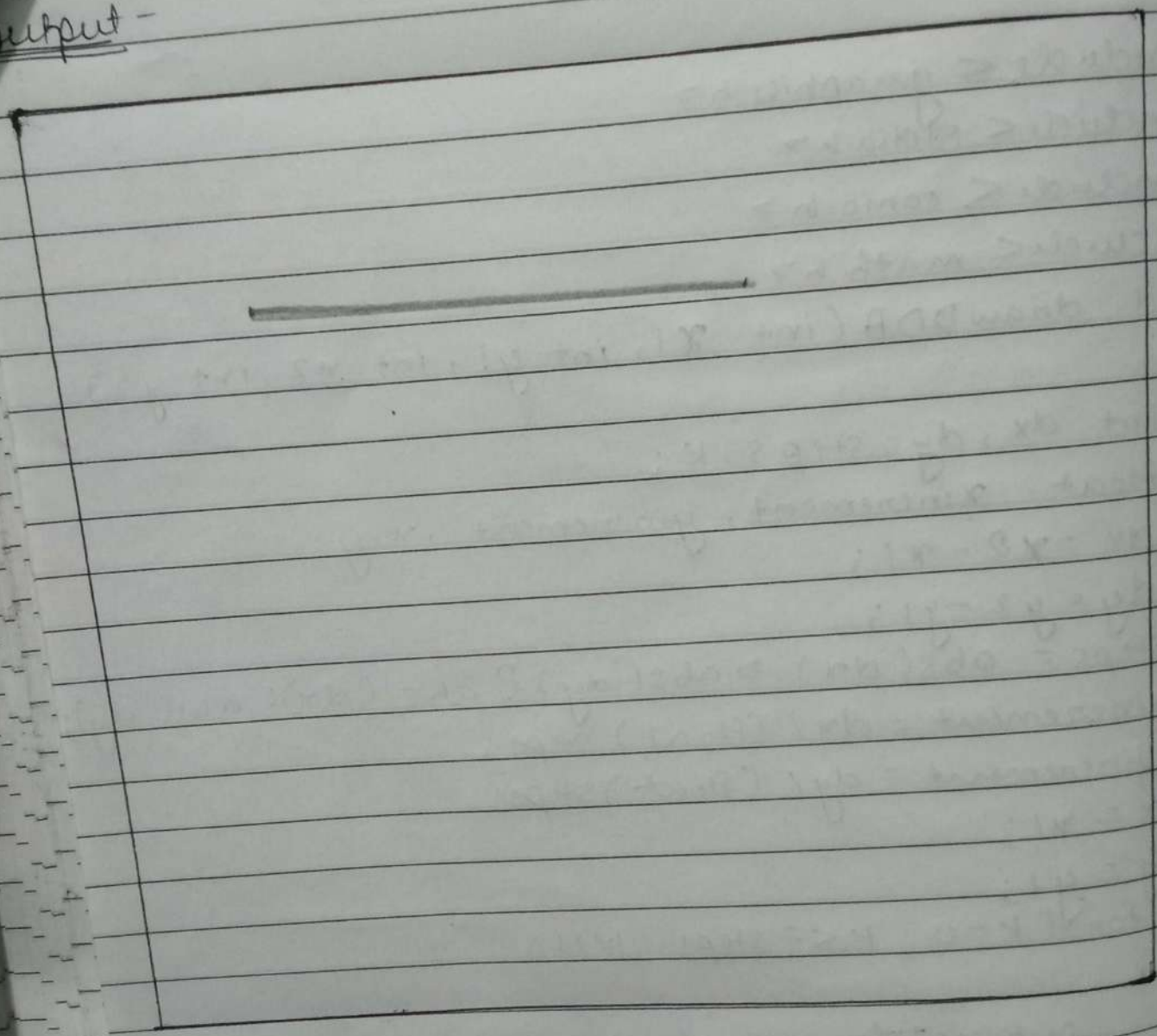
Program - 57

// WAP to implement DDA line Algorithm.

input-

```
#include <graphics.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
void drawDDA(int x1, int y1, int x2, int y2)
{
    int dx, dy, steps, k;
    float xincrement, yincrement, x, y;
    dx = x2 - x1;
    dy = y2 - y1;
    steps = abs(dx) > abs(dy) ? abs(dx): abs(dy);
    xincrement = dx/(float) steps;
    yincrement = dy/(float) steps;
    x = x1;
    y = y1;
    for(k=0; k<=steps; k++)
    {
        putpixel(random(x), random(y), WHITE);
        x += xincrement;
        y += yincrement;
        delay(10);
    }
}
void main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, " ");
    int x1=100, y1=100, x2=300, y2=300;
    drawDDA(x1, y1, x2, y2);
    getch(); closegraph(); }
```

Output -



Program - 58

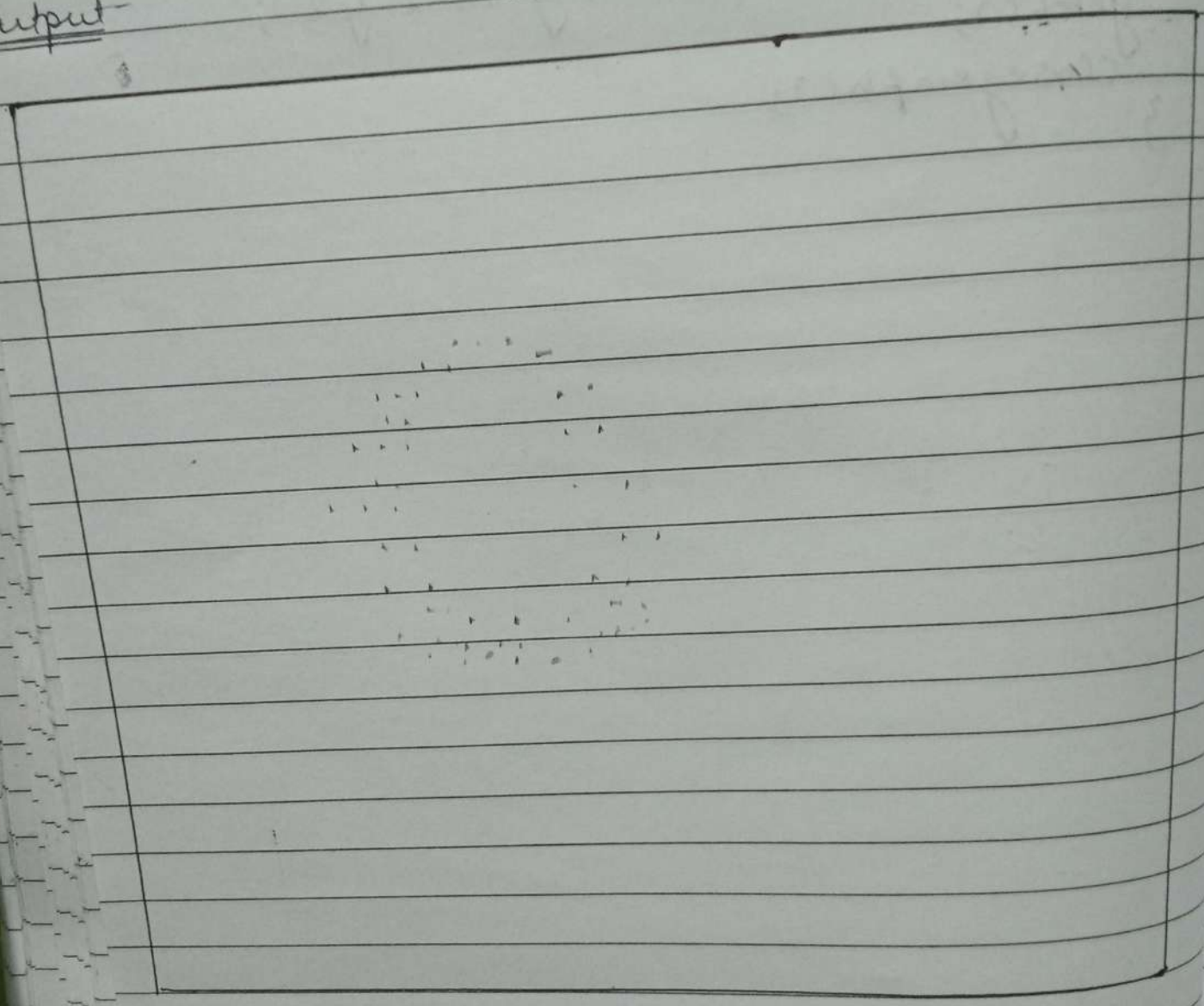
// WAP to implement Bresenhams's line Algo.

input-

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
void drawbresenhams (int x1, int y1, int x2, int y2)
{
  int dx, dy, p, x, y;
  dx = x2 - x1;
  dy = y2 - y1;
   x = x1;
   y = y1;
   p = 2 * dy - dx;
   while( x <= x2)
   {
     putpixel (x, y, WHITE);
     x++;
    if (p<0)
     p = p+2 * dy;
    else
     {
       y++;
       p = p+2 * dy - 2 * dx;
     }
    delay (10);
   }
}
void main() {
int gd= DETECT, gm;
int x1 =100, y1 = 100, x2 = 300, y2 = 100;
```

```
clrscr();
initgraph(&gd, &gm, " ");
drawbresenhams(x1, y1, x2, y2);
getch();
closegraph();
}
```

output-



Program - 59

// WAP to implement mid point circle Algorithm.

input -

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void drawcircle (int xc, int yc, int r).
{
 int x=0, y=r;
 int p=1-r;
 while (x<=y)
 {
  putpixel (xc + y, yc+y, WHITE);
  putpixel (xc+x, yc+y, WHITE);
  putpixel (xc+x, yc-y, WHITE);
  putpixel (xc-x, yc-y, WHITE);
  putpixel (xc+y, yc+x, WHITE);
  putpixel (xc-y, yc+x, WHITE);
  putpixel (xc+y, yc-x, WHITE);
  putpixel (xc-y, yc-x, WHITE);
   x++;
   if (P<0) {
     pt = 2 * x +1;
   }
   else
   {
     y--;
     pt = 2 *(x-y) +1;
   }
   delay (10);
 }
}
```

```
void main()
{
    int gd=DETECT, gm;
    clrscr();
    int xc = 200, yc = 200, radius = 100;
    initgraph(&gd, &gm, " ");
    drawcircle(xc, yc, radius);
    getch();
    closegraph();
}
```

output -

Program - 60

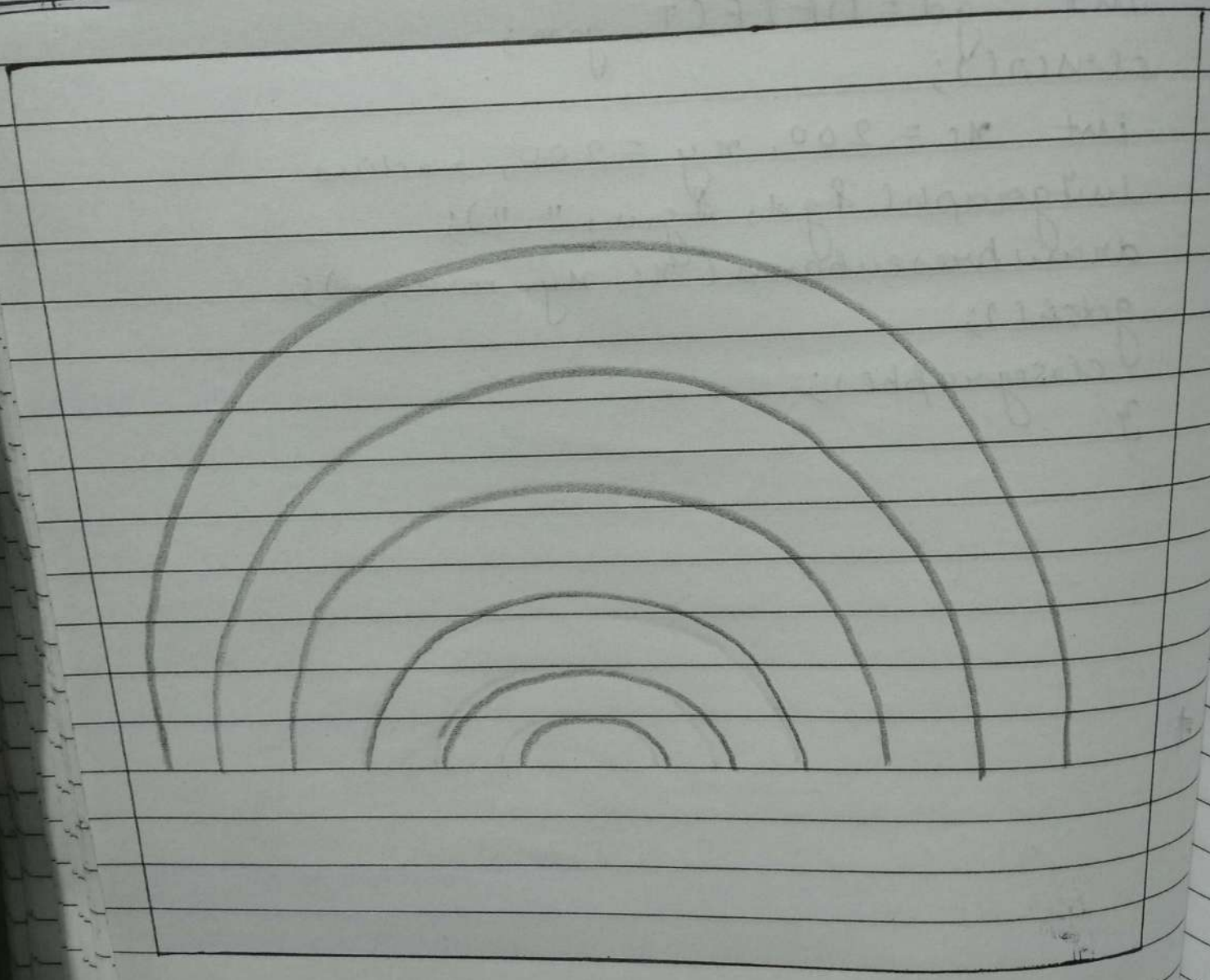
//WAP to implement Bresenham's circle Algo.

input -

```
#include< graphics.h>
#include< conio.h >
void drawbresenhams (int xc, int yc, int r)
{
  int x=0, y=r;
  int d= 3-(2 *r)
  while (x<=y)
  {
    putpixel(xc+x, yc+y, WHITE);
    putpixel(xc-x, yc+y, WHITE);
    putpixel(xc+x, yc-y, WHITE);
    putpixel(xc-x, yc-y, WHITE);
    putpixel(xc+y, yc+x, WHITE);
    putpixel(xc-y, yc-x, WHITE);
    putpixel(xc+y, yc-x, WHITE);
    putpixel(xc-y, yc-y, WHITE);
    x++;
    if(d<0)
    {
      d=d+(4 * x) +6;
    }
    else
    {
      y--;
      d=d+(4 *(x-y))+10;
    }
    delay(10);
  }
```

```
}
void main()
{
  int gd = DETECT, gm;
  clrscr();
  int xc = 200, xy = 200, radius
  initgraph(&gd, &gm, " ");
  drawbresenhams(xc, xy, radius);
  getch();
  closegraph();
}
```

Output-



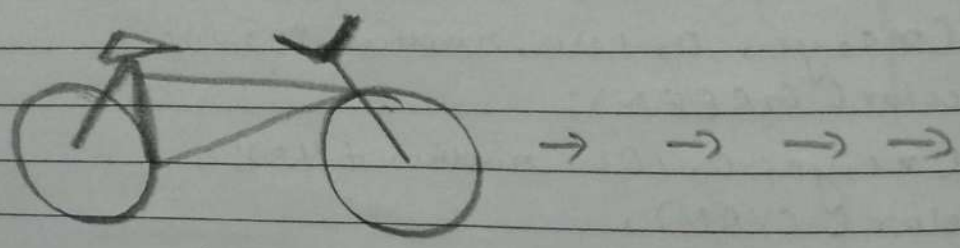
Program-61

// WAP to draw Rainbow.

input -

```
#include <graphics.h>
#include <conio.h>
void main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, " ");
    int xc = 320, yc = 240;
    int radius = 100;
    setcolor(RED);
    arc(xc, yc, 0, 180, radius + 60);
    setcolor(YELLOW);
    arc(xc, yc, 0, 180, radius + 50);
    setcolor(GREEN);
    arc(xc, yc, 0, 180, radius + 40);
    setcolor(CYAN);
    arc(xc, yc, 0, 180, radius + 30);
    setcolor(BLUE);
    arc(xc, yc, 0, 180, radius + 20);
    setcolor(MAGENTA);
    arc(xc, yc, 0, 180, radius + 10);
    getche();
    closegraph();
}
```

output-



Program - 62

// WAP to draw a moving cycle.

input -

```
#include<graphics.h>
#include<conio.h>
#include<dos.h>
void drawcycle(int x, int y)
{
    circle(x, y, 20);
    circle(x+20, y, 20);
    line(x, y, x+40, y-40);
    line(x+40, y-40, x+80, y);
    line(x+40, y-40, x+30, y-60);
    line(x+30, y-60, x+60, y-60);
    line(x+60, y-60, x+60, y);
    line(x+30, y-60, x+20, y-70);
    line(x+20, y-70, x+10, y-60);
    line(x+50, y-40, x+60, y-40);
}
void main()
{
    int gd=DETECT, gm;
    int x=50, y=300;
    initgraph(&gd, &gm, " ");
    while(!kbhit())
    {
        cleardevice();
        drawcycle(x, y)
        delay(50);
        x+=5;
        if(x>getmax())
```

```
        x = -100;
    }
}
closegraph();
getch();
}
```

Output-

DEFALT-FONT - RED
SANS-SERIF-FONT - GREEN
GOTHIC-FONT - YELLOW
SCRIPT-FONT - CYAN
SMAIL-FONT - MAGENTA

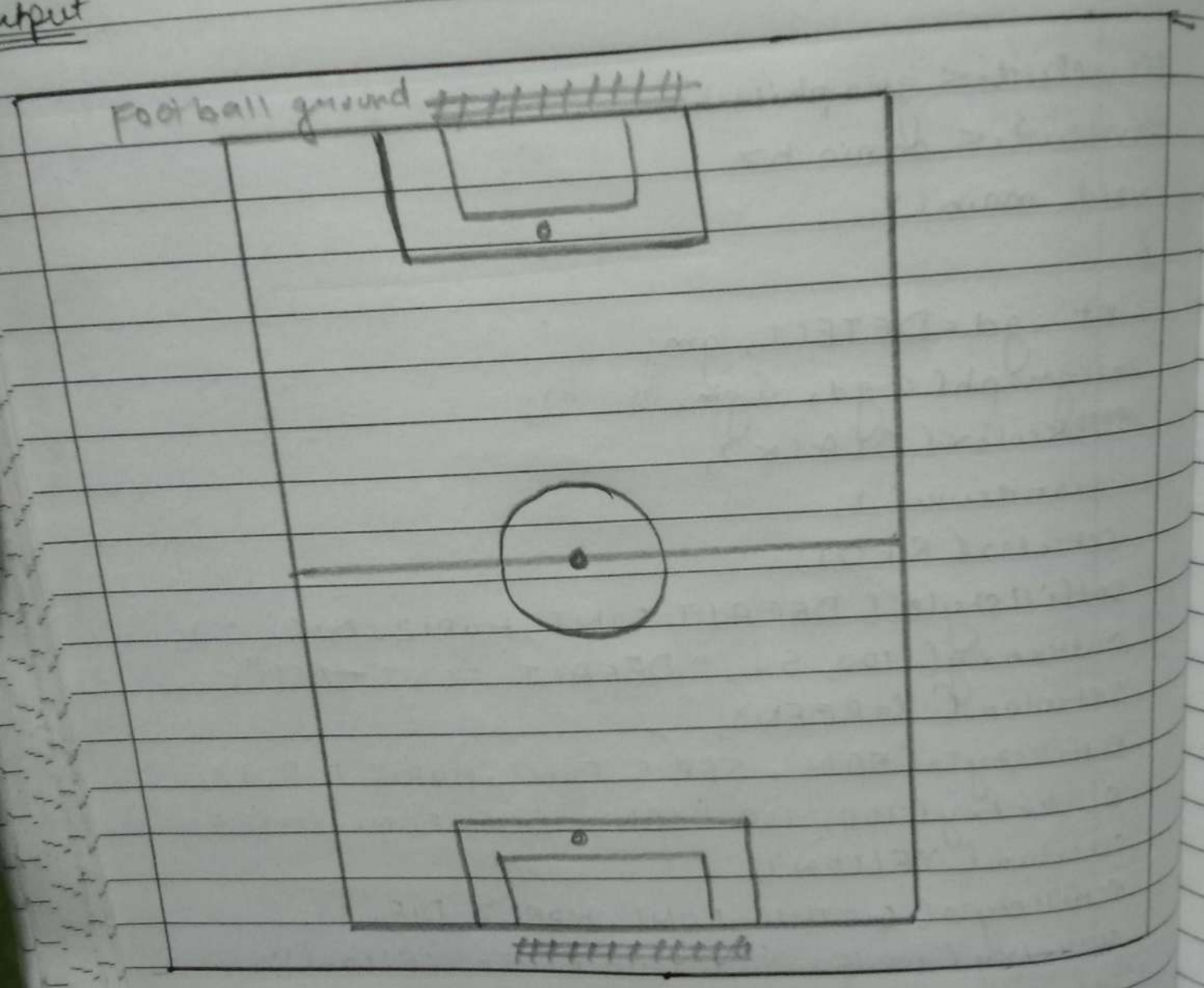
Program - 63

// WAP to use different textstyle and color

input -

```
#include< graphics.h>
#include< conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd, &gm, " ");
  setbkcolor(BLACK);
  cleardevice();
  setcolor(RED);
  setfillstyle(DEFALT_FONT, HORIZ_DIR, 2);
  outtextxy(100, 50, "DEFALT FONT - RED");
  setcolor(GREEN);
  setfillstyle(SANS_SERIF_FONT, HORIZ_DIR, 3);
  outtextxy(100, 100, "SANS-SERIF-FONT-GREEN");
  setcolor(YELLOW);
  setfillstyle(GOTHIC_FONT, HORIZ_DIR, 3);
  outtextxy(100,150, "GOTHIC-FONT- YELLOW");
  setcolor(CYAN);
  setfillstyle(SCRIPT_FONT, HORIZ_DIR, 4);
  outtextxy(100, 200, "SCRIPT_FONT - CYAN");
  setcolor(MAGENTA);
  setfillstyle(SMALL_FONT, HORIZ_DIR, 3);
  outtextxy(100, 250, "SMALL_FONT");
  getch();
  closegraph();
}
```

utput



Program-64

// WAP to draw Football ground

input-

```
#include< graphics.h>
#include< conio.h>
void main()
{
  int gd=DETECT, gm;
  initgraph(&gd,&gm," ");
  setcolor(WHITE);
  setbkcolor(GREEN);
  rectangle(50,40,650,450);
  rectangle(50,140,150,340);
  line(330,40,330,450);
  rectangle(50,170,400,300);
  rectangle(500,440,600,340);
  rectangle(550,170,600,300);
  circle(330,240,50);
  pieslice(330,240,0,360,5);
  pieslice(125,240,0,360,5);
  pieslice(525,240,0,360,5);
  ellipse(150,240,270,90,30,45);
  ellipse(500,240,90,270,30,45);
  setfillstyle(HATCH-FILL,WHITE);
  bar(25,200,50,270);
  bar(600,200,625,270);
  setcolor(RED);
  setfillstyle(10,HORIZ-DIR,1);
  outtextxy(200,2"Football ground");
  getch();
  closegraph();
}
```

Output-

Different line style

SOLID-LINE

DOTTED-LINE

CENER-LINE

DASHED-LINE

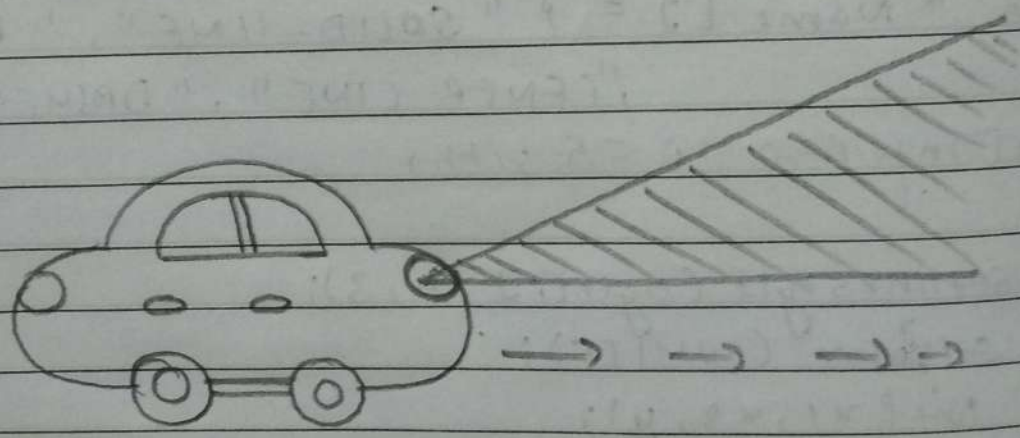
Program-65

// WAP to illustrate different line style

input-

```
#include< graphics.h>
#include< conio.h>
void drawlinestyle()
{
  int x1 = 100, x2 = 400, y = 50;
  int style [] = { SOLID_LINE, DOTTED_LINE,
                   CENER_LINE, DASHED_LINE};
  char *Name [] = { "SOLID_LINE", "DOTTED_LINE
                   ,"CENER LINE", "DASHED_LINE");
  for( int i=0 ; i <5 ; i++)
  {
    setlinestyle (style[i], 0, 3);
    setcolor (WHITE);
    line( x1, x2, y);
    setcolor (YELLOW);
    outtextxy ( x2+ 20, y-5, Name [i])
      y+=40;
  }
}
void main()
{
    int gd=DETECT, gm;
    initgraph( &gd, &gm, " ");
    drawlinestyle();
    getch();
    closegraph();
}
```

Output -



Program - 66

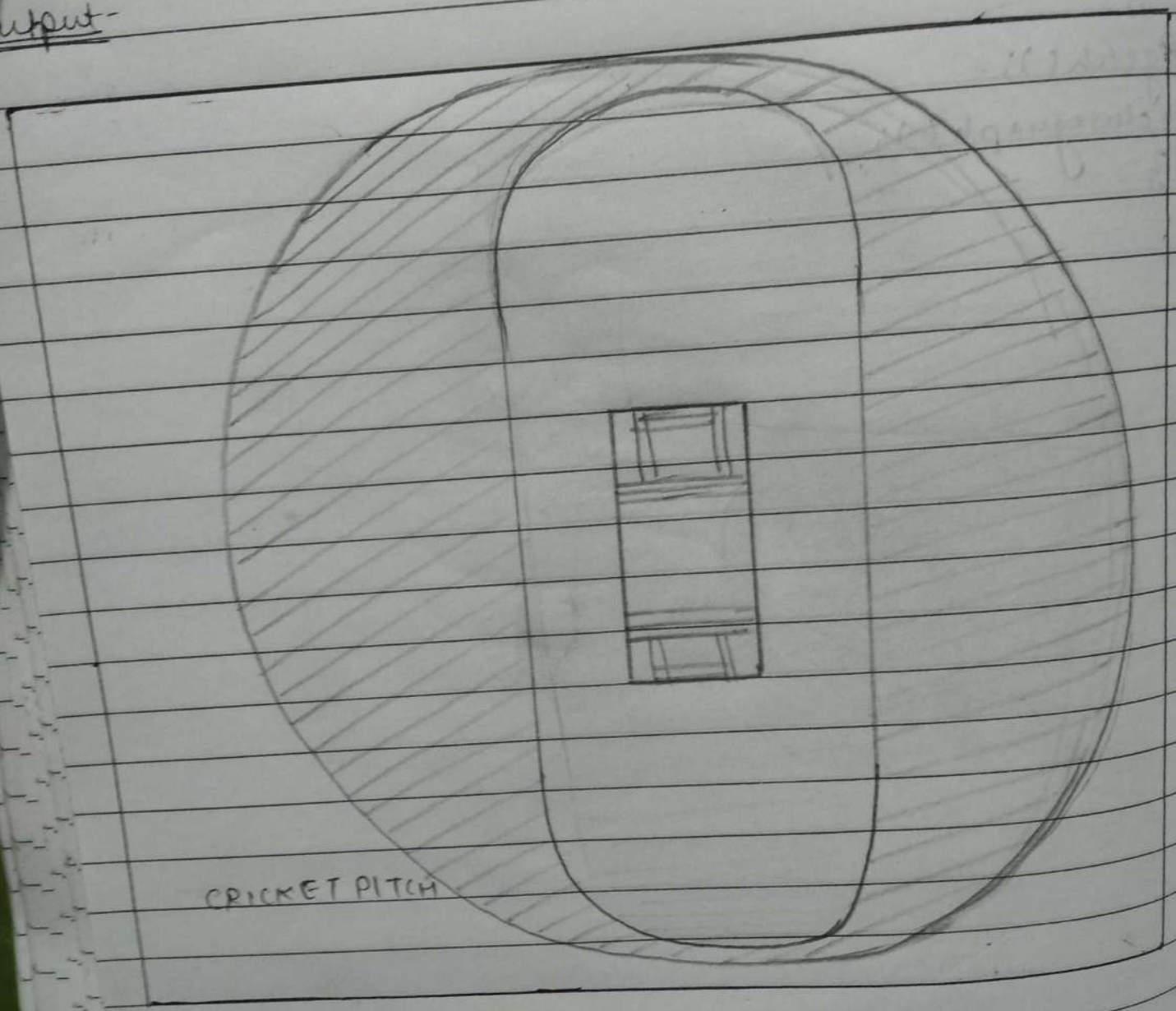
// WAP to draw moving car

input -

```
#include <graphics.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
void main()
{
int gd= DETECT, gm;
initgraph( &gd, &gm, " ");
for( int i= 100, i <= 550; i++)
{
ellipse (100+ 50+i , 300, 20, 160, 80, 70);
ellipse( 175+50+i , 317, 205, 90, 50, 40);
arc( 76+i , 315, 90, 270, 30);
ellipse (100+ 50+ i , 290, 20, 160, 50, 40);
line( 52+ 50+ i , 277, 148+ 50+ i, 277);
rectangle( 99+50+ i , 250, 101+ 50+i , 277);
circle( 215+i , 355, 20);
circle( 215+i , 355, 8);
circle( 100+i , 355, 20);
circle(100+i, 355, 8);
rectangle( 195+ i , 352 , 120+i , 355);
ellipse( 160+i , 300, 0, 360, 7, 3);
ellipse( 110 +i, 300, 0 , 360, 7, 3 );
ellipse( 260+i , 300, 90, 340, 14, 11);
ellipse ( 50+ i, 300, 218, 80, 14, 16);
line( 260 + i , 300, getmaxx( ), 180);
line( 260+ i, 300, getmaxy( ), 300);
setfillstyle( 6, YELLOW);
```

```
floodfill (300 + i, 295, WHITE);
 delay (20);
 cleardevice();
}
getch();
closegraph();
}
```

output-



Program-67

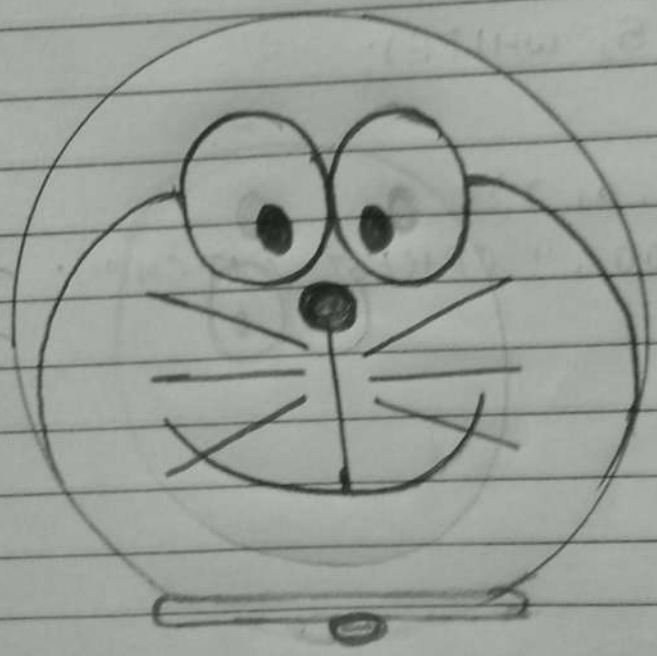
// WAP to draw cricket pitch

input-

```
#include<graphics.h>
#include<conio.h>
void main()
{
 int gd=DETECT, gm;
 initgraph(&gd, &gm, " ");
 setcolor(WHITE);
 bar(290, 170, 350, 173);
 bar(290, 287, 350, 290);
 bar(300, 150, 301, 170);
 bar(340, 150, 339, 170);
 bar(300, 310, 301, 290);
 bar(340, 310, 339, 290);
 bar(300, 155, 339, 156);
 bar(300, 304, 339, 305);
 rectangle (290, 150, 350, 310);
 line(230, 129, 230, 331);
 line(410, 129, 410, 331);
 arc(320, 130, 0, 180, 90);
 arc(320, 330, 180, 360, 90);
 setfillstyle(SOLID-FILL, GREEN);
 floodfill(320, 330, WHITE);
 setfillstyle(SOLID-FILL, 12);
 floodfill(292, 151, WHITE);
 floodfill(305, 152, WHITE);
 floodfill(345, 152, WHITE);
 floodfill(305, 160, WHITE);
 floodfill(305, 190, WHITE);
```

```
floodfill( 305, 300, WHITE);
floodfill( 345, 308, WHITE);
floodfill( 330, 308, WHITE);
floodfill ( 295, 308, WHITE);
circle ( getmaxx( ) / 2, getmaxy( ) / 2, 240);
setfillstyle ( 6, GREEN);
floodfill( 300, 5, WHITE);
setcolor (14);
settextstyle ( 1, 0, 2);
outtextxy (0, 400, " CRICKET PITCH");
getch();
closegraph();
}
```

output -



DOREMON

Program - 68

// WAP to draw cartoon character

input -

```
// DOREMON
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int gd=DETECT, gm;
initgraph(&gd, &gm, " ");
setcolor(WHITE);
outtextxy(400, 400, "DOREMON");
ellipse(290, 200, 0, 360, 19, 30);
ellipse(330, 200, 0, 360, 19, 30);
setfillstyle(SOLID-FILL, WHITE);
floodfill(290, 200, WHITE);
floodfill(330, 200, WHITE);
setcolor(BLACK);
setfillstyle(SOLID-FILL, BLACK);
fillellipse(295, 210, 5, 10);
fillellipse(325, 210, 5, 10);
floodfill(295, 210, BLACK);
floodfill(325, 210, BLACK);
setcolor(WHITE);
circle(309, 235, 9);
setfillstyle(SOLID-FILL, RED);
floodfill(309, 235, WHITE);
line(309, 245, 309, 300);
arc(309, 253, 200, 340, 50);
line(295, 249, 260, 235);
line(295, 257, 260, 257);
```

```
line( 295, 264, 260, 274);
line( 324, 249, 360, 235);
line( 324, 257, 360, 257);
line( 324, 264, 360, 274);
setcolor( WHITE);
ellipse( 270, 275, 90, 220, 33, 80);
setfillstyle( SOLID_FILL, BLUE);
floodfill( 270, 195, WHITE);
ellipse( 350, 275, 320, 90, 33, 80);
ellipse( 310, 255, 310, 230, 80, 120);
setcolor( WHITE);
setfillstyle( SOLID_FILL, RED);
line( 250, 345, 369, 345);
line( 250, 355, 369, 355);
line( 250, 345, 250, 355);
line( 369, 345, 369, 355);
floodfill( 255, 350, WHITE);
setcolor( WHITE);
setfillstyle( SOLID_FILL, YELLOW);
circle( 305, 360, 10);
floodfill( 305, 360, WHITE);
getch();
closegraph();
}
```