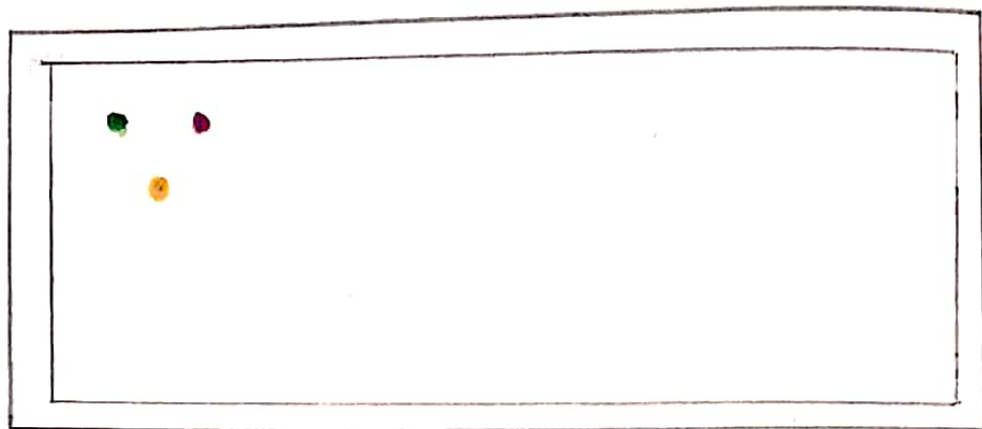




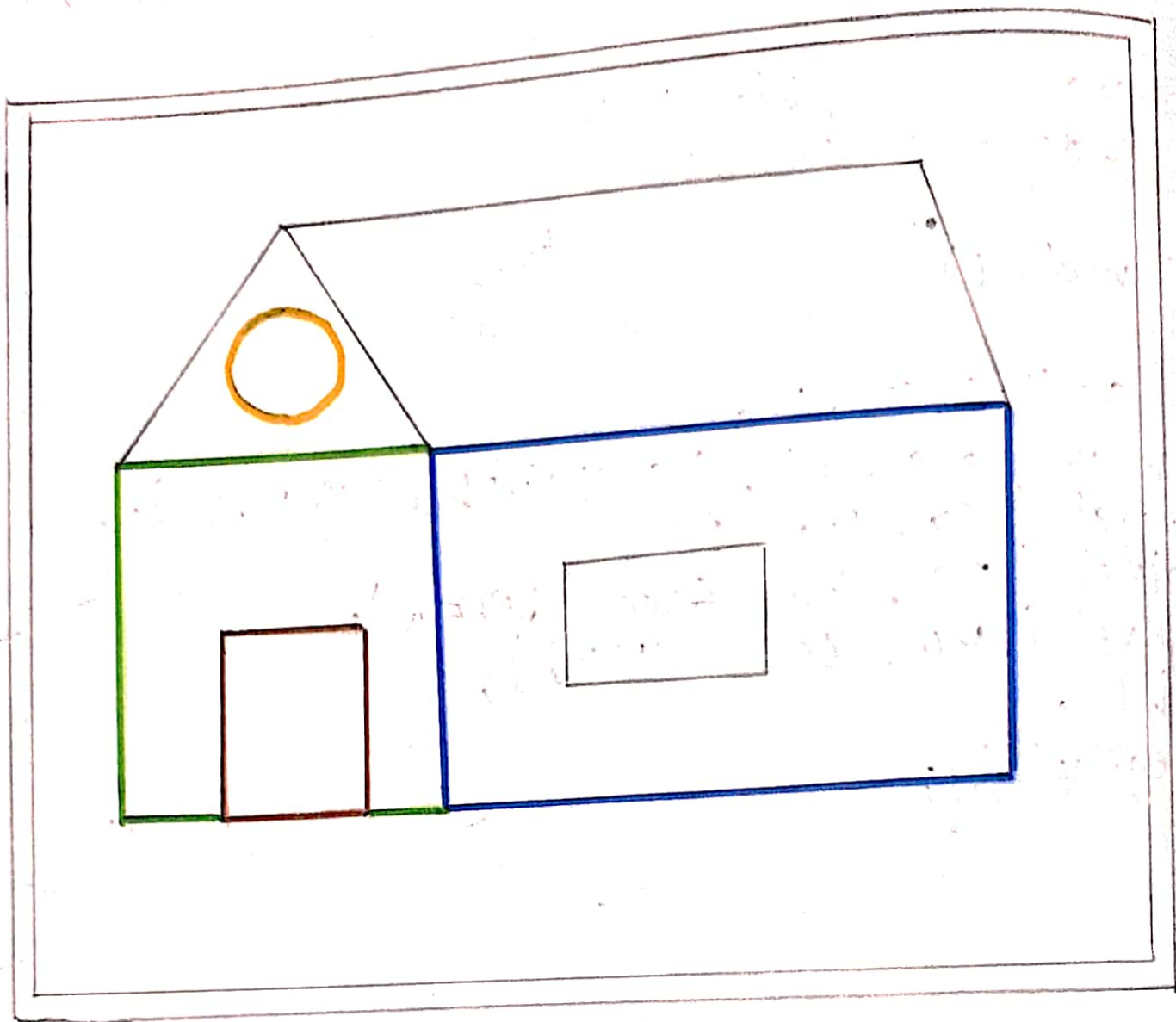
1 Write a program to plot a pixel on screen.

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

```
void main()
```

```
{
    int a = DETECT, b;
    initgraph (&a, &b, "c:\\turbo3\\BGI");
    putpixel (50, 40, RED);
    putpixel (30, 40, Green GREEN);
    putpixel (40, 60, YELLOW);
    getch();
    closegraph();
}
```

Teacher's Signature : _____



2 Write a program to draw a house using line and rectangle inbuilt function

```
#include <stdio.h>
#include <graphics.h>
void main ()
{
    int gd = DETECT, gm;
    initgraph (&gd, &gm, "C:\\turborc3\\BGI");
    setcolor (10);
    rectangle (150, 100, 250, 250);
    setcolor (9);
    rectangle (250, 100, 420, 250);
    setcolor (6);
    rectangle (190, 180, 230, 250);
    setcolor (7);
    rectangle (340, 150, 280, 200);
    line (200, 50, 150, 100);
    line (200, 50, 250, 100);
    line (385, 50, 420, 100);
    line (200, 50, 385, 50);
    setcolor (7);
    circle (200, 80, 17);
    getch();
    closegraph ();
}
```

Teacher's Signature : _____

Enter the value of X_1 and Y_1 ;
100 100

Enter the value of X_2 and Y_2 ;
100 200

100 is steps and 0.000000 and
1.000000 are X_{incr} and Y_{incr}



3 Write a program to draw a line using DDA algorithm between two given end Points.

```
#include <stdio.h>
#include <graphics.h>
#include <math.h>
#include <stdlib.h>
#define ROUND(a) ((int) (a + .5))
```

```
void main()
```

```
{
    int gd, gm;
    float x, y, xincr, yincr;
    int i, x1, y1, x2, y2, dx, dy, steps;
    printf("Enter the value of x1 and y1 :");
    scanf("%d %d", &x1, &y1);
    printf("Enter the value of x2 and y2 :");
    scanf("%d %d", &x2, &y2);
    dx = x2 - x1;
    dy = y2 - y1;
    x = x1;
    y = y1;
    if (abs(dx) > abs(dy))
        steps = abs(dx);
    else
        steps = abs(dy);
    xincr = dx / (float) steps;
    yincr = dy / (float) steps;
```

Teacher's Signature : _____

```
printf ("%d is steps and %.f and %.f are  
Xincr and Yincr", steps, Xincr, Yincr);  
getch();  
gd = DETECT;  
initgraph (&gd, &gm, "C:\\turbo3\\BGI");  
putpixel (x, y, 14RED);  
for (i = 0; i < steps; i++)  
{  
    x = x + Xincr;  
    y = y + Yincr;  
    putpixel (ROUND(x), ROUND(y), 14RED);  
    delay (20);  
}  
getch();  
closegraph();  
}
```

Teacher's Signature : _____

Enter First Point : 100 100

Enter Second Point : 200 150



4 Write a program to draw line using Bresenham's algorithm.

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

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

```
void drawline (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)
```

```
    {  
        if (p >= 0)
```

```
        {  
            putpixel (x, y, 7);
```

```
            delay (100);
```

```
            y = y + 1;
```

```
            p = p + 2 * dy - 2 * dx;
```

```
        }  
        else
```

```
        {  
            putpixel (x, y, 7);
```

```
            delay (100);
```

```
            p = p + 2 * dy;
```

Teacher's Signature : _____

```
X = X + 1;
```

```
}  
}  
int main()
```

```
{  
    int gd = DETECT, gm, x1, x2, y1, y2;  
    initgraph (&gd, &gm, "C:\\turbo3\\BGI");
```

```
    printf("Enter First Point:");  
    scanf("%d %d", &x1, &y1);
```

```
    printf("Enter Second Point:");  
    scanf("%d %d", &x2, &y2);
```

```
    cleardevice();
```

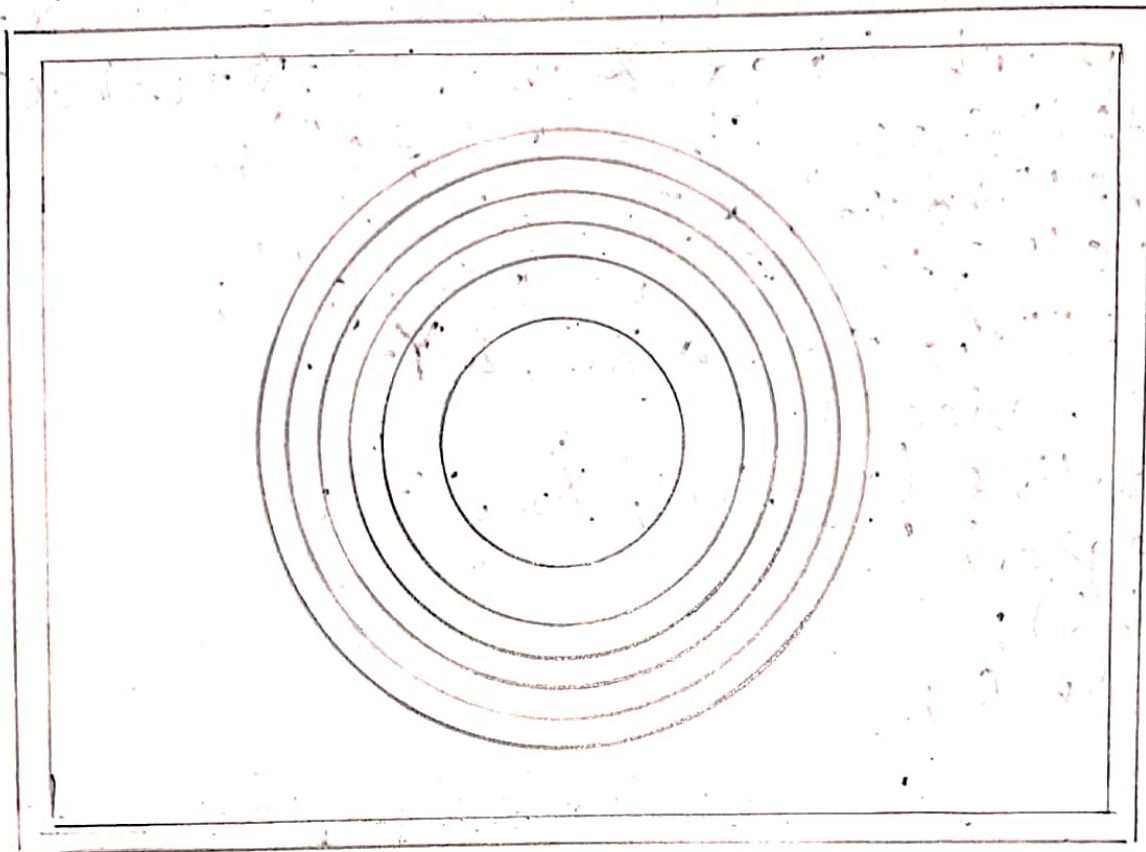
```
    drawline (x1, y1, x2, y2);
```

```
    closegraph();
```

```
    getch();
```

```
    return 0;  
}
```

Teacher's Signature : _____



5 Write a program to draw concentric circles on screen.

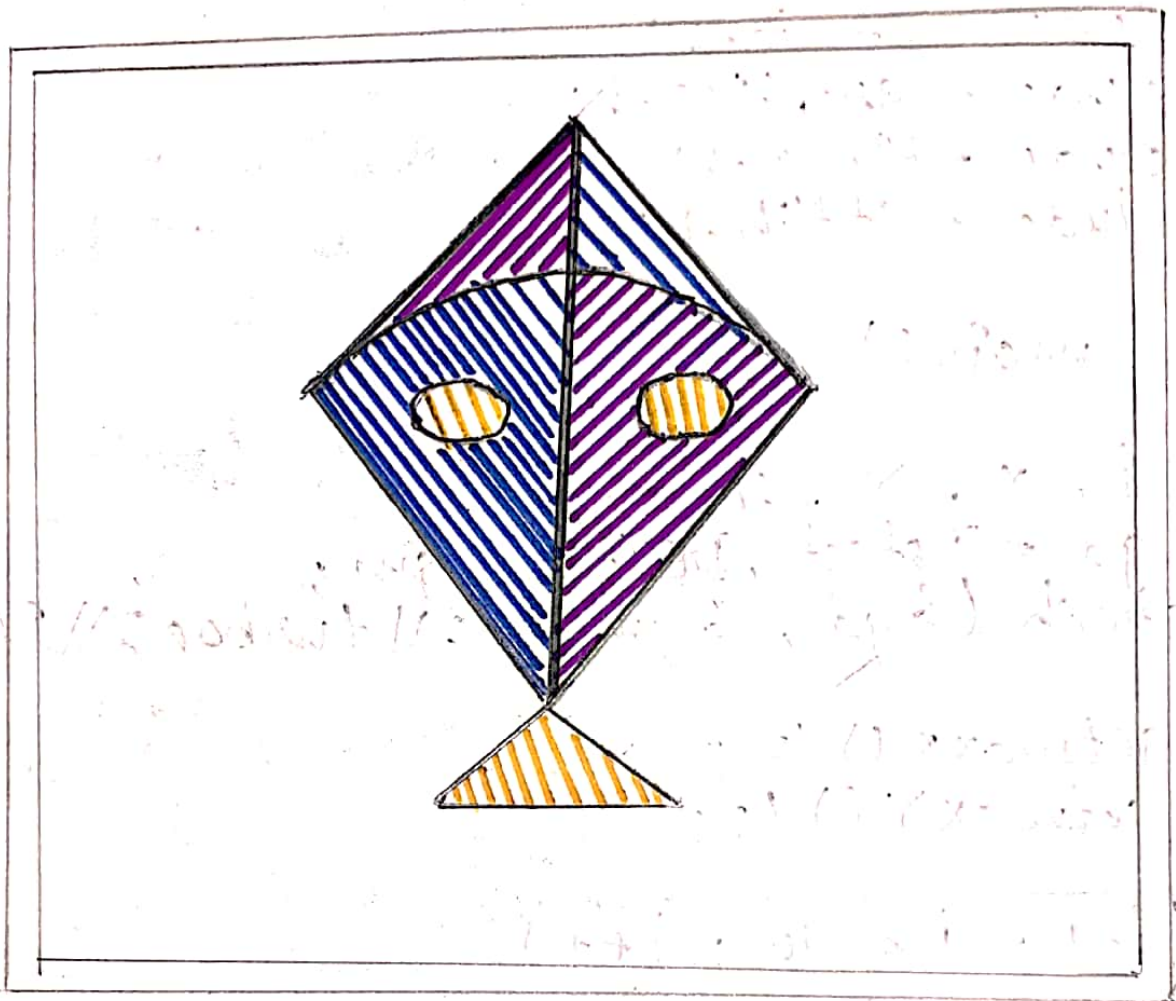
```
#include <stdio.h>
#include <graphics.h>
#include <time.h>
#include <stdlib.h>

void main()
{
    int i, x, y, r = 10;
    int gd = DETECT, DETECT, gm;
    initgraph(&gd, &gm, "C:\\turbo3\\BGI");

    x = getmaxx() / 2;
    y = getmaxy() / 2;

    for (i = 1; i <= 10; i++)
    {
        setcolor(random(16));
        circle(x, y, r);
        r = r + 10;
        delay(50);
    }

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



6 Write a program to draw a patterned Kite (Use Fillpoly and setFillstyle)

```
#include <stdio.h>
#include <graphics.h>
void main()
{
    int x = DETECT, y;
    int a[8] = {300, 300, 350, 350, 230, 350, 300, 300};
    initgraph (&x, &y, "C:\\turbo3\\BGI");
    line (200, 200, 300, 100);
    line (300, 100, 400, 200);
    line (400, 200, 300, 300);
    line (300, 100, 300, 300);
    line (300, 300, 200, 200);
    arc(300, 300, 45, 135, 140);

    setFillstyle (5, 1);
    Floodfill (301, 105, WHITE);

    setFillstyle (4, 5);
    Floodfill (299, 105, WHITE);

    setFillstyle (5, 1);
    Floodfill (299, 275, WHITE);

    setFillstyle (4, 5);
    Floodfill (301, 275, WHITE);

    setFillstyle (5, 14);
```

Teacher's Signature : _____

Fillellipse (260, 210, 18, WHITE);

setfillstyle (5, 14);

Fillellipse (340, 210, 18, WHITE);

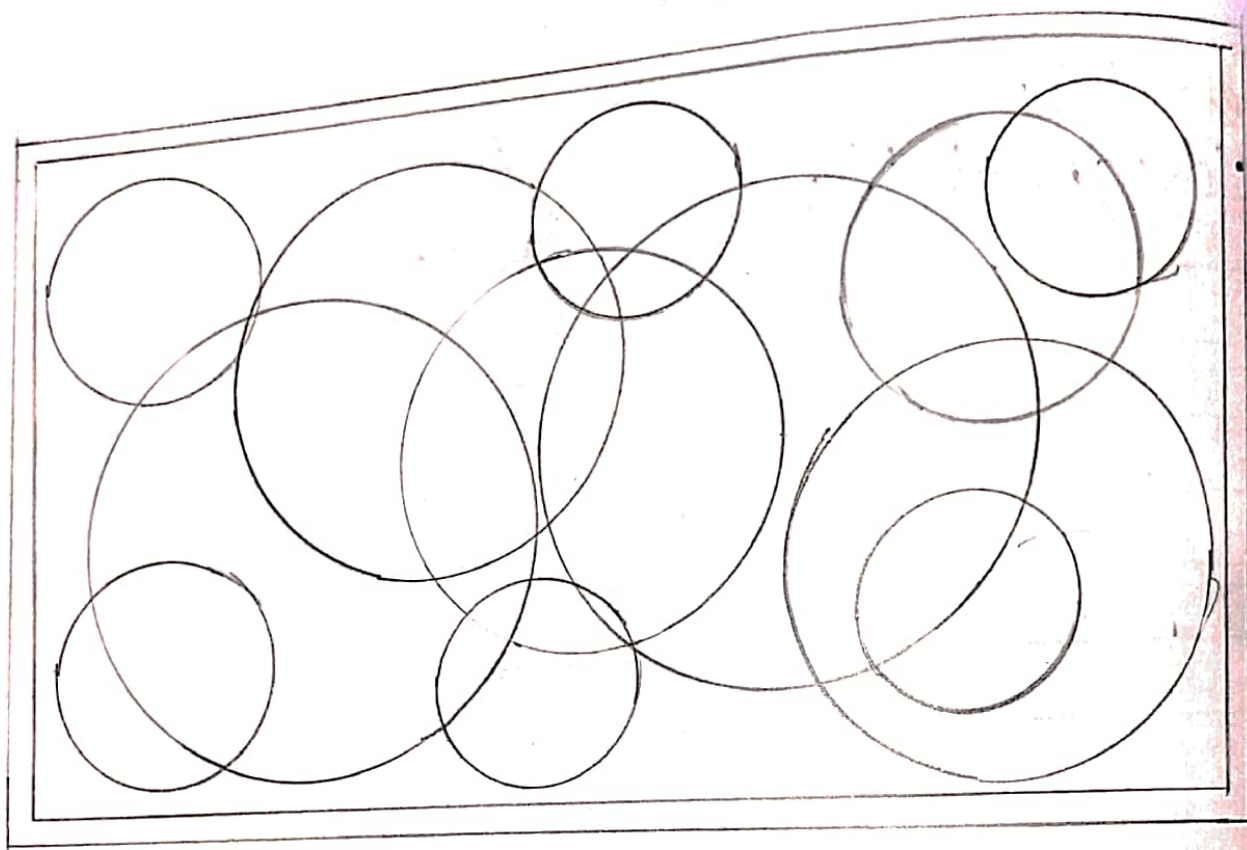
drawpoly (4, a);

fillpoly (4, a);

getch();

} closegraph();
}

Teacher's Signature : _____



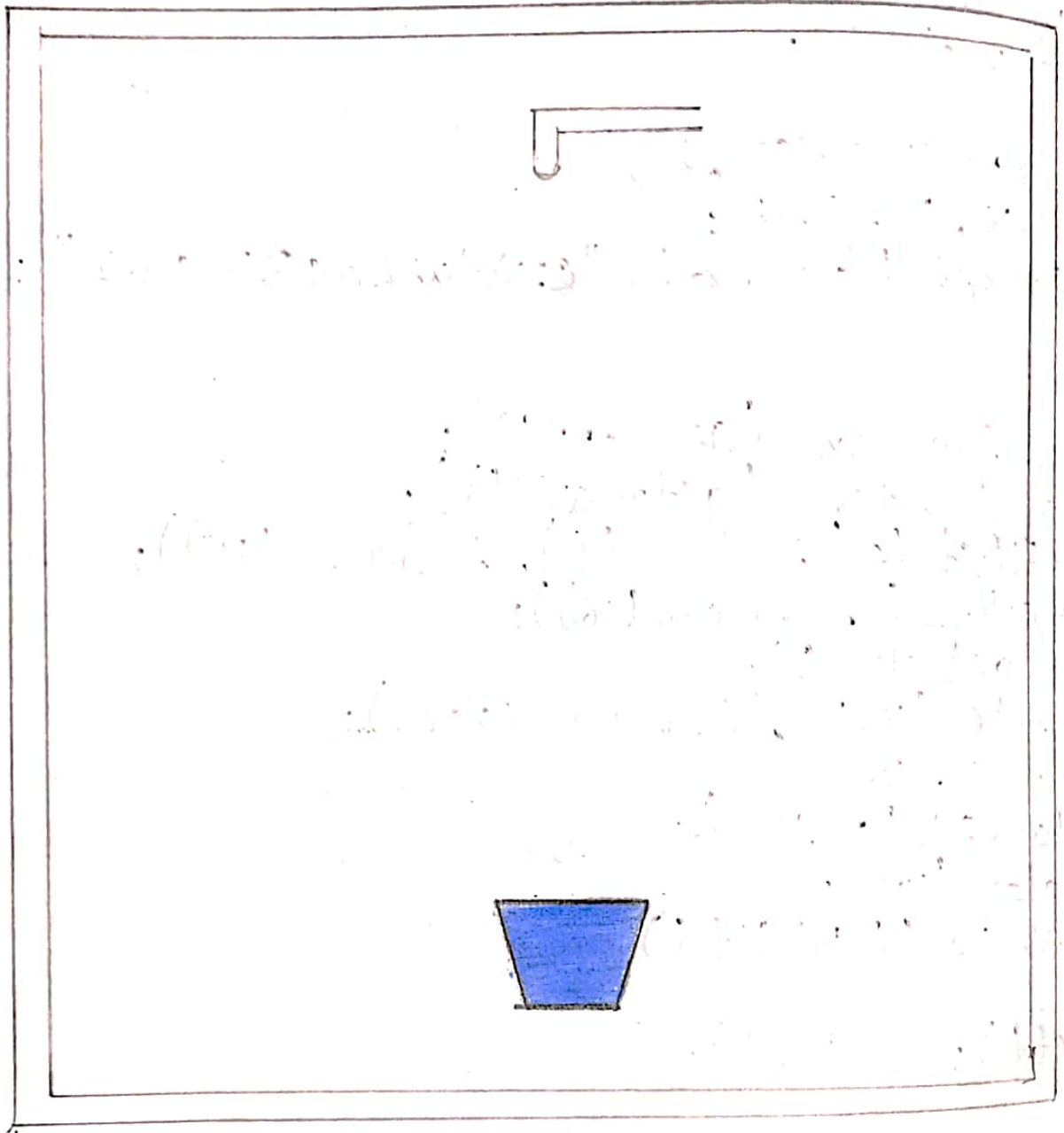
7 Write a Program to create Screen Saver.

```
#include <stdio.h>
#include <graphics.h>
#include <stdlib.h>
void main ()
{
    int a = DETECT, b;
    int x, y, c, r;
    initgraph (&a, &b, "c:\\turbo3\\BGI");
    do
    {
        x = random (getmaxx());
        y = random (getmaxy());
        setfillstyle (random (11), random (15));
        setcolor (random (16));
        c = getcolor ();
        circle (x, y, random (200));
        r = r + 10;
        floodfill (x, y, c);
        delay (100);
    } while (!kbhit());

    getch(); getch();

    closegraph ();
}
```

Teacher's Signature : _____



8 Write a program to implement boundary fill algorithm (Fill bucket with tap water)

```
#include <stdio.h>
#include <graphics.h>
void tap()
{
    line (getmaxx()/2-10, 50, getmaxx()/2+60, 50);
    line (getmaxx()/2, 60, getmaxx()/2+60, 60);
    line (getmaxx()/2-10, 50, getmaxx()/2-10, 70);
    line (getmaxx()/2, 60, getmaxx()/2, 70);
    arc (getmaxx()/2-5, 70, 180, 360, 5);
}
void drop (int c, int d)
{
    int Y = 100, i;
    setfillstyle (c, d);
    for (i=1; i<=6; i++)
    {
        fillellipse (getmaxx()/2, Y, 5, 10);
        delay (50);
        Y = Y + 50;
    }
}
void bucket()
{
    line (getmaxx()/2-30, 380, getmaxx()/2+30, 380);
    line (getmaxx()/2-20, 420, getmaxx()/2+20, 420);
    line (getmaxx()/2-30, 380, getmaxx()/2-20, 420);
}
```

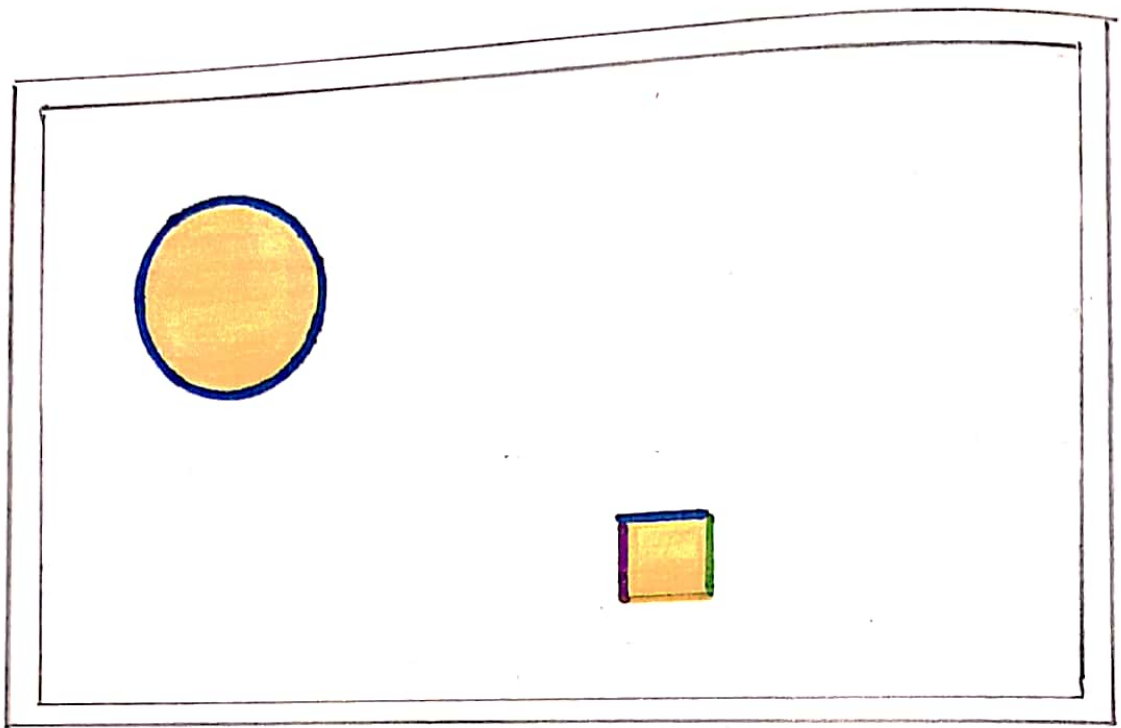
Teacher's Signature : _____

```
line(getmaxx()/2+30, 380, getmaxx()/2+20, 420);  
void bf(int x, int y, int fill-color, int boundary-color)  
{  
    if(getpixel(x, y) != boundary-color && getpixel(x, y) != fill-color)  
    {  
        putpixel(x, y, fill-color);  
        delay(5);  
        bf(x+1, y, fill-color, boundary-color);  
        bf(x, y+1, fill-color, boundary-color);  
        bf(x-1, y, fill-color, boundary-color);  
        bf(x, y-1, fill-color, boundary-color);  
    }  
}  
int main()  
{  
    int a = DETECT, b;  
    initgraph(&a, &b, "C:\\turbo c3\\BGI");  
    tap();  
    bucket();  
    while(!kbhit());  
    drop(1, BLUE);  
    bf(getmaxx()/2-19, 419, BLUE, WHITE);  
    delay(1000);  
    setcolor(BLACK);  
    drop(0, BLACK);  
    getch();  
}
```

Teacher's Signature : _____

```
getch();  
close graph();  
return 0;  
}
```

Teacher's Signature : _____



9 Write a program to implement floodfill algorithm

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

```
void Flood (int x, int y, int fillcolor, int defaultcolor)
{
    if (getpixel (x, y) == defaultcolor)
    {
        putpixel (x, y, fillcolor);
        delay (5);
        Flood (x+1, y, fillcolor, defaultcolor);
        Flood (x, y+1, fillcolor, defaultcolor);
        Flood (x-1, y, fillcolor, defaultcolor);
        Flood (x, y-1, fillcolor, defaultcolor);
    }
}
```

```
int main ()
```

```
{
    int gd = DETECT, gm;
    int x = 100, y = 100, r = 14;
    initgraph (&gd, &gm, "C:\\turbo\\3\\BGI");
    setcolor (1);
    circle (100, 100, 20);
    Flood (100, 100, 14, 0);
}
```

```
setcolor (1);
```

Teacher's Signature : _____

```
line (200, 200, 220, 200);
```

```
setcolor(2);
```

```
line (220, 200, 220, 220);
```

```
setcolor(14);
```

```
line (220, 220, 200, 220);
```

```
setcolor(5);
```

```
line (200, 220, 200, 200);
```

```
Flood(210, 210, 14, 0);
```

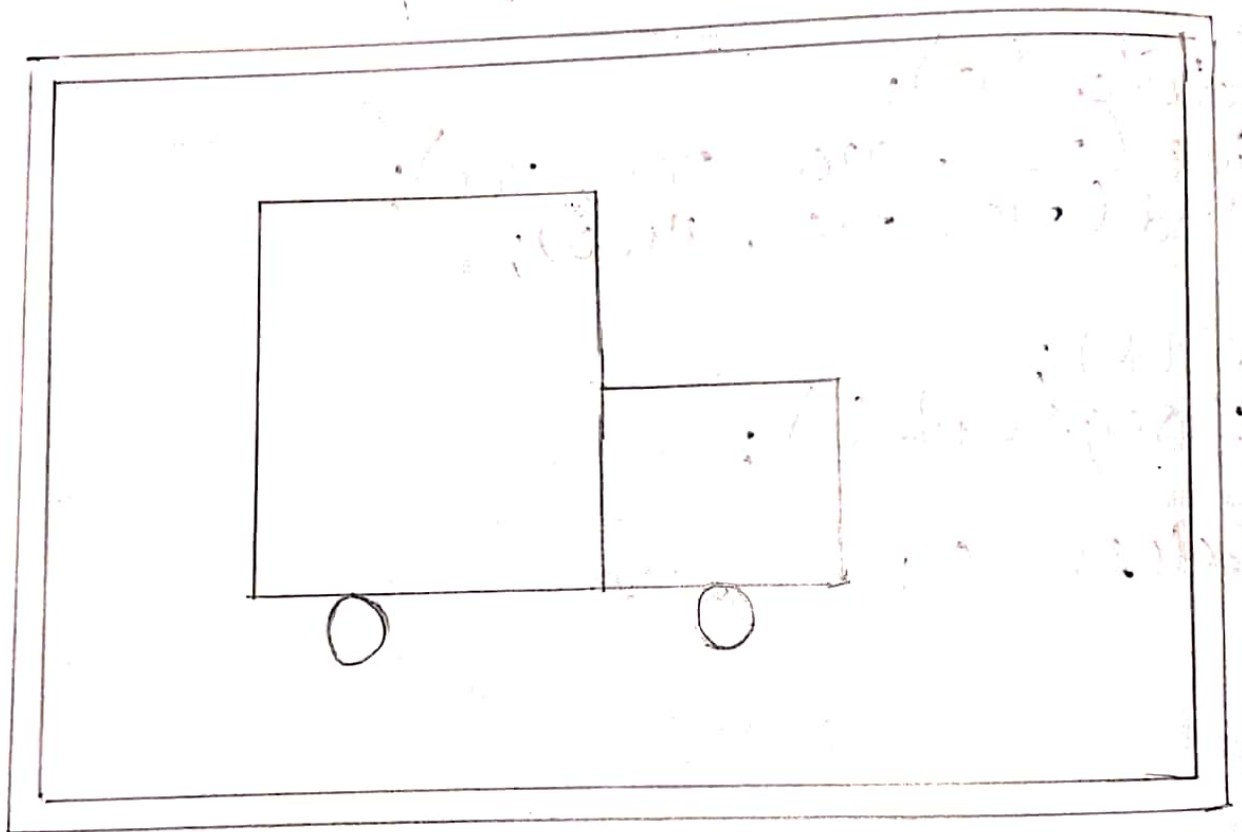
```
getch();
```

```
closegraph();
```

```
return 0;
```

```
}
```

Teacher's Signature : _____



10 Write a program to show moving car.

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

```
void car (int i)
```

```
{
    rectangle (50+i, 275, 150+i, 400);
    rectangle (150+i, 350, 200+i, 400);
    circle (75+i, 410, 10);
    circle (175+i, 410, 10);
}
```

```
int main()
```

```
{
    int x, i, gd = DETECT, gm;
```

```
    initgraph (&gd, &gm, "C:\\turbo\\3\\BGI")
```

```
do
```

```
{
    cleardevice();
```

```
    setcolor (WHITE);
```

```
    car (x);
```

```
    delay (100);
```

```
    x = x + 5;
```

```
    if (x >= getmaxx() - 150)
```

```
    x = 0;
```

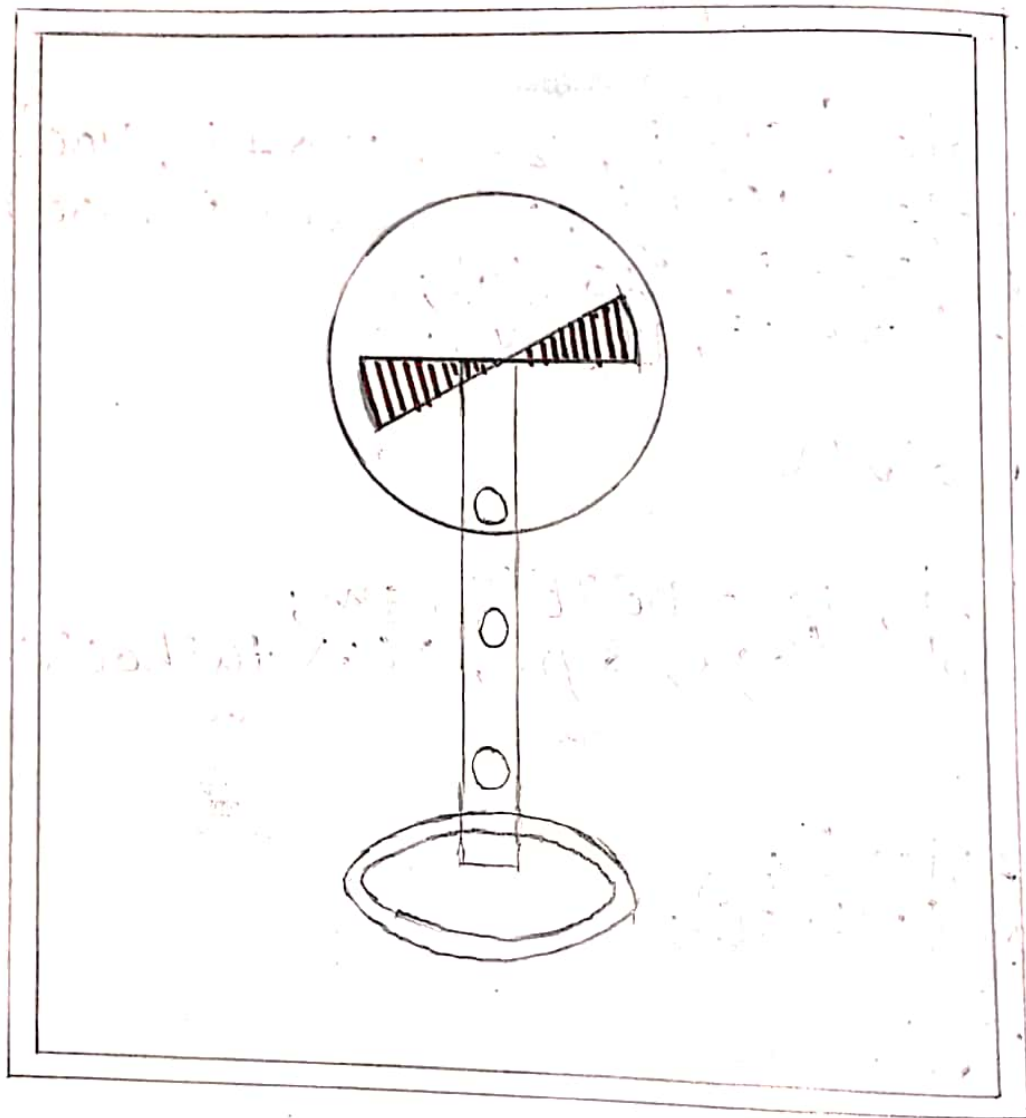
```
} while (!kbhit());
```

```
getch();
```

```
closegraph();
```

```
return 0;
```

Teacher's Signature : _____



11 Write a program to show moving Fan blades

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

```
Void blade()
```

```
{
    SetFillstyle (6,6);
    pieslice (200, 200, 0, 50, 50);
    pieslice (200, 200, 180, 225, 50);
}
```

```
Void blade2()
```

```
{
    SetFillstyle (6,6);
    pieslice (200, 200, 90, 90+60, 50);
    pieslice (200, 200, 270, 270+60, 50);
}
```

```
Void blade3()
```

```
{
    SetFillstyle (6,6);
    pieslice (200, 200, 180, 180+50, 50);
    pieslice (200, 200, 360, 360+50, 50);
}
```

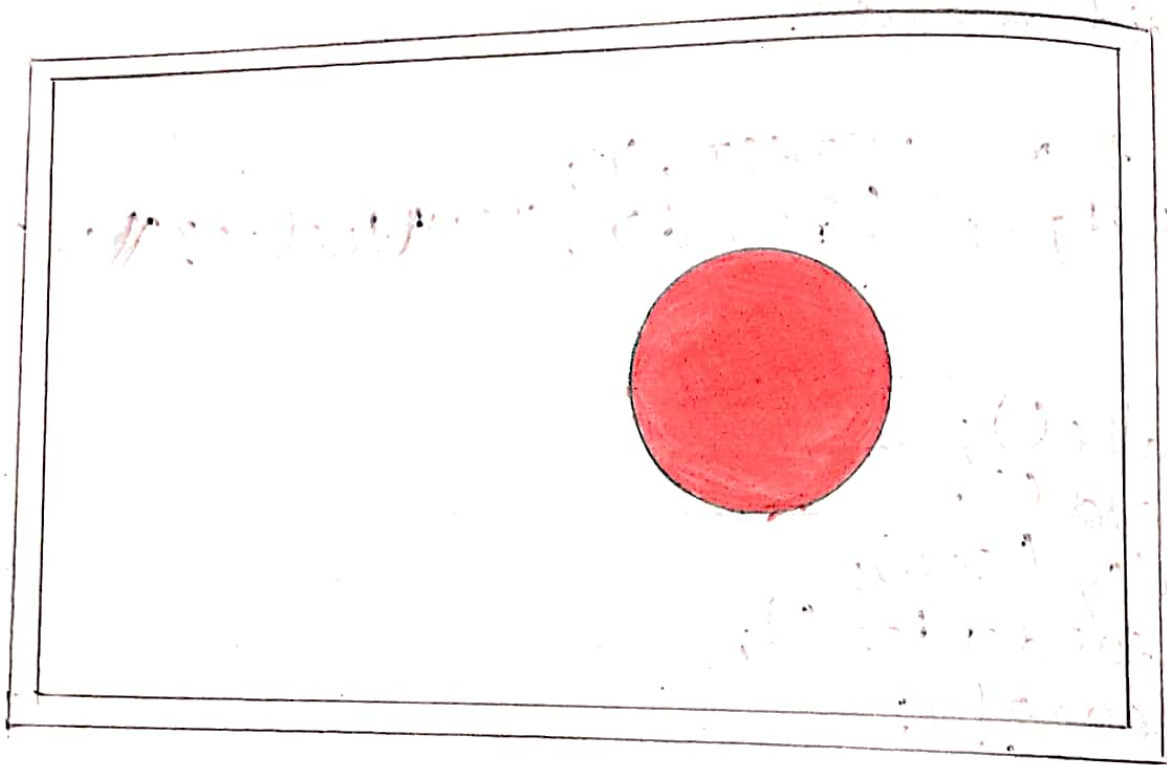
```
Void stand()
```

```
{
    SetFillstyle (9,8);
    rectangle (190, 200, 210, 350);
}
```

Teacher's Signature : _____

```
circle(200, 200, 60);
circle(200, 250, 5);
circle(200, 280, 5);
circle(200, 315, 5);
ellipse(200, 350, 190, 185, 60, 20);
ellipse(200, 350, 190, 185, 70, 20);
}
void main()
{
    int x = DETECT, y;
    initgraph(&x, &y, "C:\\turbo3\\BGI");
do
{
    blade();
    stand();
    delay(500);
    cleardevice();
    blade2();
    stand();
    delay(500);
    cleardevice();
    blade3();
    stand();
    delay(500);
    cleardevice();
} while (!kbhit());
getch();
closegraph();
}
```

Teacher's Signature : _____



12 Write a program to show use of arrow keys.

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

void main()
{
    int a = DETECT, b, x, y, r = 40;
    char ch;
    initgraph(&a, &b, "C:\\turbox3\\bgi");
    x = getmaxx() / 2;
    y = getmaxy() / 2;
    setfillstyle(1, RED);
    circle(x, y, r);
    floodfill(x, y, WHITE);
do
{
    ch = getch();
    switch(ch)
    {
        case 'K':
            cleardevice();
            x = x - 10;
            circle(x, y, r);
            floodfill(x, y, WHITE);
            if (x <= 40)
                x = getmaxx() - 40;
            break;
    }
}
```

Teacher's Signature : _____

Case 'H':

```
cleardevice();  
Y = Y - 10;  
circle(X, Y, R);  
FloodFill(X, Y, WHITE);  
if(Y <= 40);  
    Y = getmaxy() - 40;
```

break;

Case 'M':

```
cleardevice();  
X = X + 10;  
circle(X, Y, R);  
FloodFill(X, Y, WHITE);  
if(X >= getmaxx() - 40)  
    X = 40;
```

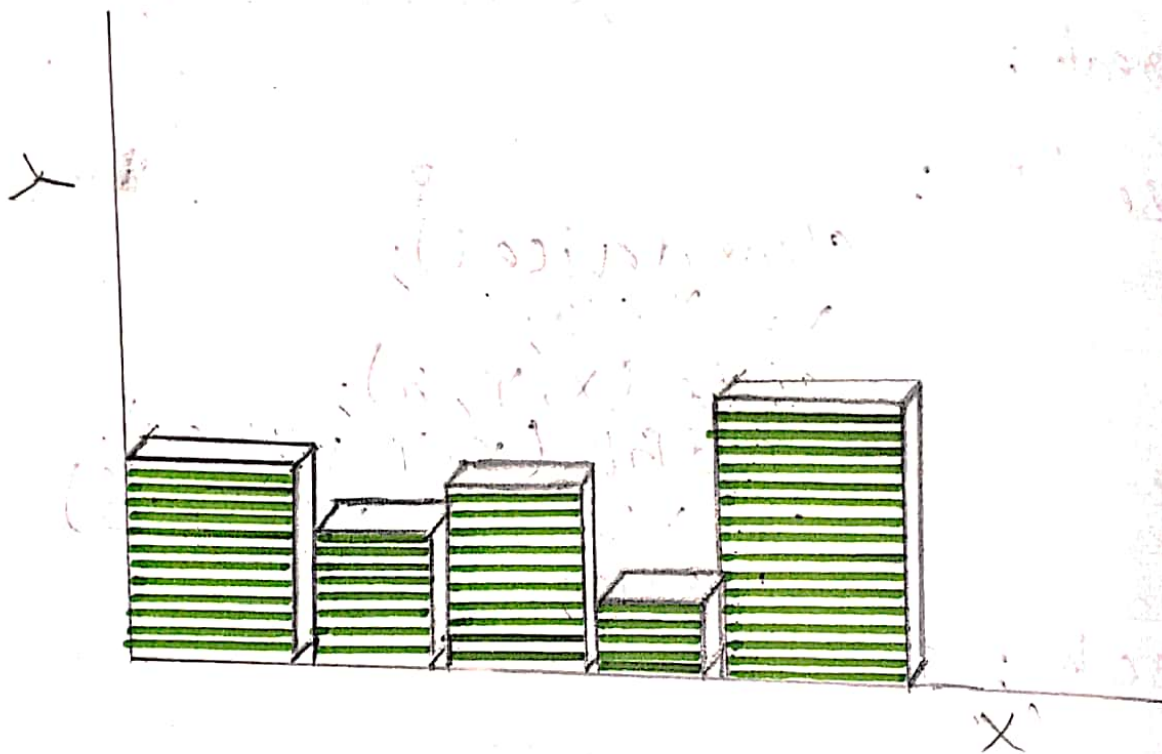
break;

Case 'P':

```
cleardevice();  
Y = Y + 10;  
circle(X, Y, R);  
FloodFill(X, Y, WHITE);  
if(Y >= getmaxy() - 40)  
    Y = 0;
```

```
}  
while(ch != 13);  
getch();  
}
```

Teacher's Signature : _____



B Write Program to draw 3d bar graph on basis of input marks.

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

```
Void main()
```

```
{
    int a, b;
```

```
    int m[5] = {80, 60, 70, 30, 100};
    a = DETECT;
```

```
    initgraph(&a, &b, "c:\\turbo3\\bgi");
    setfillstyle(2, 10);
```

```
    line(50, 50, 50, getmaxy()-50);
```

```
    line(50, getmaxy()-50, getmaxx()-50,
        getmaxy()-50);
```

```
    Outtextxy(10, getmaxx()-10, "X");
```

```
    settextstyle(0, 1, 1);
```

```
    Outtextxy(20, 50, "Y");
```

```
    bar3d(50, getmaxy()-50-m[0], 50+50,
        getmaxy()-50, 10, 1);
```

```
    bar3d(110, getmaxy()-50-m[1], 110+50,
        getmaxy()-50, 10, 1);
```

```
    bar3d(170, getmaxy()-50-m[2], 170+50,
        getmaxy()-50, 10, 1);
```

Teacher's Signature : _____

```
bar3d (230, getmaxy() - 50 - m[3], 230 + 50,  
getmaxy() - 50, 10, 1);
```

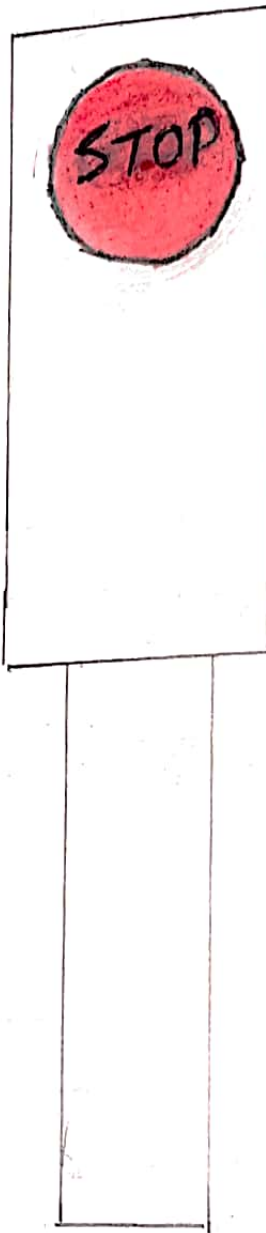
```
bar3d (290, getmaxy() - 50 - m[4], 290 + 50,  
getmaxy() - 50, 10, 1);
```

```
getch();
```

```
closegraph();
```

```
}
```

Teacher's Signature : _____



14 Write a program to show traffic light simulation.

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

```
void red()
```

```
{
    circle(75, 75, 20);
    setfillstyle(1, RED);
    floodfill(60, 70, WHITE);
    outtextxy(60, 70, "STOP");
}
```

```
void yellow()
```

```
{
    circle(75, 125, 20);
    setfillstyle(1, YELLOW);
    floodfill(70, 120, WHITE);
    outtextxy(60, 120, "READY");
}
```

```
void green()
```

```
{
    circle(75, 175, 20);
    setfillstyle(1, GREEN);
    floodfill(75, 170, WHITE);
    outtextxy(60, 170, "GO");
}
```

Teacher's Signature : _____

```
void Stand()
{
    rectangle (50, 50, 100, 200);
    rectangle (60, 350, 90, 200);
}

void main()
{
    int a = DETECT, b;
    initgraph (&a, &b, "c:\\turbo3\\bgi");
    do
    {
        Stand();

        red();
        delay (1000);
        cleardevice ();

        Stand();

        Yellow();
        delay (1000);
        cleardevice ();

        Stand();

        green();
        delay (1000);
```

Teacher's Signature : _____

```
.clear device ();
```

```
}
```

```
while (!kbhit());
```

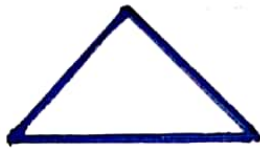
```
getch();
```

```
close graph();
```

```
}
```

Teacher's Signature : _____

Enter the Points : 100 100 150 150 50 50



1. Translation
2. Rotation
3. Scalling
4. Exit

Enter Your choice : 1

Enter the translation Point: 100 100



Enter Your choice : 4

Exit.

15 Write a program to show basic transformations.

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

void main()
{
    int gd = DETECT, gm;
    int x1, x2, x3, y1, y2, y3, c;
    int nx1, nx2, nx3, ny1, ny2, ny3;
    int sx, sy, xt, yt, r;
    float t;
    initgraph(&gd, &gm, "c:\\turboc3\\bgi");
    printf("\n Enter the points:");
    setcolor(BLUE);
    scanf("%d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3);
    line(x1, y1, x2, y2);
    line(x2, y2, x3, y3);
    line(x3, y3, x1, y1);
    getch();
    printf("\n 1. Translation\n 2. Rotation\n 3. Scalling\n 4. Exit\n");
    do
    {
        printf("Enter Your choice:");
        scanf("%d", &c);
    }
```

Teacher's Signature : _____

```

Set color (RED);
Switch (c);
{
  class1 :
  printf ("In Enter the translation Point:");
  scanf ("%d %d", &xt, &yt);

```

```

  nx1 = x1 + xt;          ny1 = y1 + yt;
  nx2 = x2 + xt;          ny2 = y2 + yt;
  nx3 = x3 + xt;          ny3 = y3 + yt;

```

```

  line (nx1, ny1, nx2, ny2);
  line (nx2, ny2, nx3, ny3);
  line (nx3, ny3, nx1, ny1);

```

```

  getch();
  break;

```

```

Case 2:
  printf ("In Enter the angle of rotation:");
  scanf ("%d", &r);
  t = (3.14 * r) / 180;

```

```

  nx1 = abs (x1 * cos (t) - y1 * sin (t));
  ny1 = abs (x1 * sin (t) - y1 * cos (t));
  nx2 = abs (x2 * cos (t) - y2 * sin (t));
  ny2 = abs (x2 * sin (t) - y2 * cos (t));
  nx3 = abs (x3 * cos (t) - y3 * sin (t));

```

Teacher's Signature : _____

```
nY3 = abs (x3 * sin(t) - Y3 * cos(t));
```

```
line (nX1, nY1, nX2, nY2);
```

```
line (nX2, nY2, nX3, nY3);
```

```
line (nX3, nY3, nX1, nY1);
```

```
getch();  
break;
```

Case 3;

```
printf("Enter the angle of scaling:");  
scanf("%d %d", &SX, &SY);
```

```
nX1 = X1 * SX;
```

```
nY1 = Y1 * SY;
```

```
nX2 = X2 * SX;
```

```
nY2 = Y2 * SY;
```

```
nX3 = X3 * SX;
```

```
nY3 = Y3 * SY;
```

```
line (nX1, nY1, nX2, nY2);
```

```
line (nX2, nY2, nX3, nY3);
```

```
line (nX3, nY3, nX1, nY1);
```

```
getch();  
break;
```

Case 4;

```
printf("Exit");
```

```
exit(0);
```

```
break; } }
```

Teacher's Signature : _____

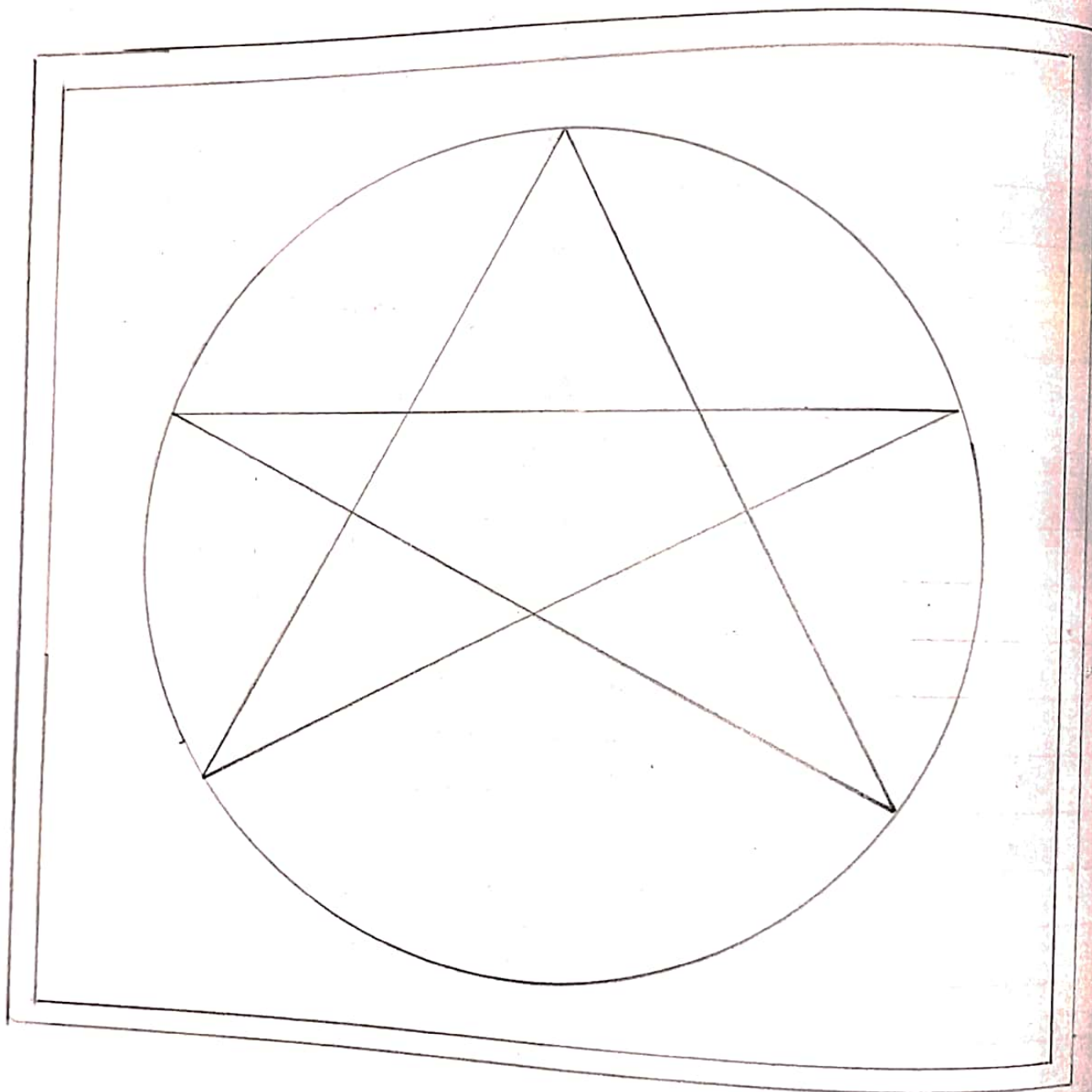
Expt. No.

Date

Page No.

```
while (!kbhit());  
getch();  
closegraph();  
}
```

Teacher's Signature : _____



16 Write a program to draw a star connecting 5 points on circles circumference.

```
#include <graphics.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

```
void main()
```

```
{
    int gd, gm;
    int i;
    int xp, yp;    int pt[10];
    float theta, angle = 0;
    int a = 0;
    int r = 150;
    gd = DETECT;
    initgraph(&gd, &gm, "C:\\turbo\\3\\BGI");
    x = getmaxx() / 2;
    y = getmaxy() / 2;
    circle(x, y, r);
    for(i = 0, a = 0; i <= 360; i = i + 72, a = a + 2)
    {
        theta = (i * 3.14) / 180;
        pt[a] = (int)(x + r * cos(theta));
        pt[a + 1] = (int)(y + r * sin(theta));
        delay(50);
    }
```

Teacher's Signature : _____

```
cleardevice();  
putpixel(pt[a], pt[a+1], GREEN);  
delay(5);  
}  
line(pt[8], pt[9], pt[2], pt[3]);  
line(pt[2], pt[3], pt[6], pt[7]);  
line(pt[6], pt[7], pt[0], pt[1]);  
line(pt[0], pt[1], pt[4], pt[5]);  
line(pt[4], pt[5], pt[8], pt[9]);  
  
getch();  
  
closegraph();  
  
}
```

Teacher's Signature : _____