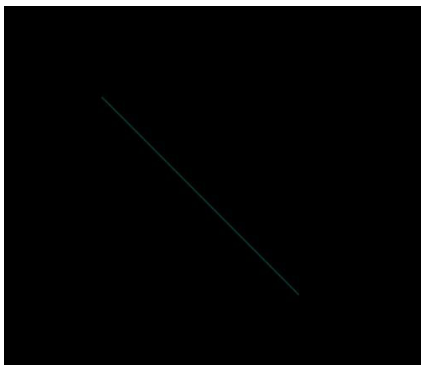


7. Write a program to draw a line using DDA.

CODE

```
#include <stdio.h>
#include <graphics.h>
#include <math.h>
#include <conio.h> int
main() {
    int gdriver = DETECT, gmode;
    initgraph(&gdriver, &gmode, (char*)"C:\\MinGW\\lib\\libbgi.a");
    int x1, y1, x2, y2, dx, dy, steps, xinc, yinc;
    printf("Enter value of x1,y1 and x2,y2");
    scanf("%d%d%d%d", &x1, &y1, &x2, &y2);
    dx = x2-x1;    dy =
y2-y1;    if(abs(dx) >
abs(dy)){        steps =
abs(dx);    } else {
steps = abs(dy);
    }    xinc = dx/steps;
yinc = dy/steps;    for(int
i=0; i<steps; i++){
putpixel(x1,y1,3);    x1
= x1 + xinc;    y1 = y1 +
yinc;    delay(100);
    }    getch();
closegraph();
return 0;
}
```

OUTPUT

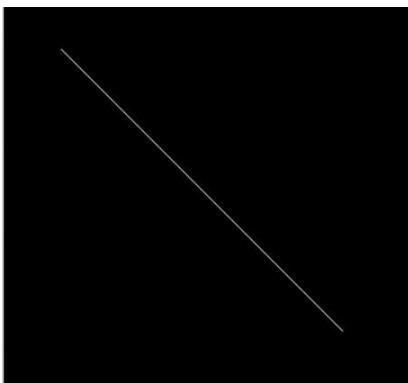


8. WAP to draw a line using Bresenham's Line drawing algorithm.

CODE

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h> #include<math.h> int main() { float
dy,dx,x,y,x1,x2,y1,y2,p,gdriver = DETECT, gmode;
initgraph(&gdriver, &gmode, (char*)"C:\\MinGW\\lib\\libbgi.a");
printf("\n Enter x1, y1 co-ordinate and x2, y2 co-ordinate : ");
scanf("%f %f %f %f",&x1,&y1,&x2,&y2); dx=abs(x2-x1);
dy=abs(y2-y1); if(x1<x2) { x=x1; y=y1;
}
if(x1>x2) {
x=x2; y=y2;
}
p=(2*dy)-dx;
while(x<=x1 || x<=x2){
if(p<0){ x=x+1;
y=y; p=p+(2*dy);
} else{ x=x+1;
y=y+1;
p=p+(2*dy)-(2*dx);
}
putpixel((int)x,(int)y,WHITE);
} getch();
closegraph();
return 0;
}
```

OUTPUT



9. WAP to draw a circle using Mid Point shortcut algorithm.

CODE

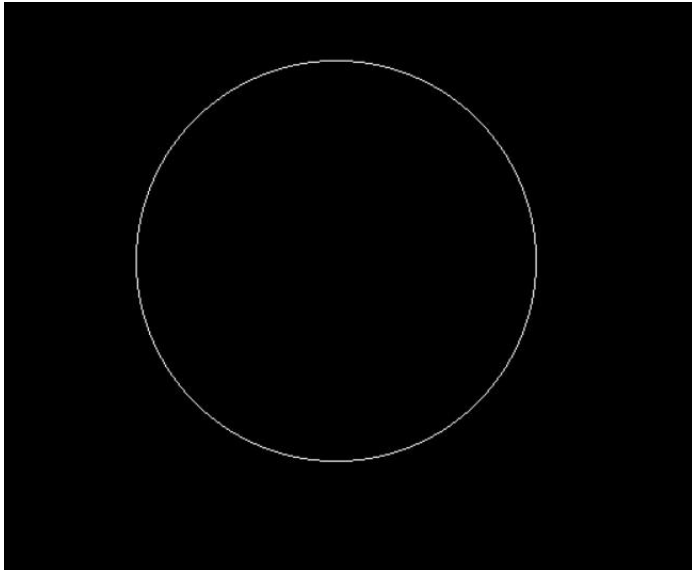
```
#include <stdio.h>
#include <graphics.h>
```

```
void drawCircle(int xc, int yc, int x, int 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 - x, WHITE);  
}
```

```
void midpointCircle(int xc, int yc, int r) {  
    int x = 0;    int y = r;  
    int p = 1 - r;  
  
    while (x <= y) {  
        drawCircle(xc, yc, x, y);  
        x++;    if (p < 0) {  
            p = p + 2 * x + 1;  
        } else {  
            y--;  
            p = p + 2 * x - 2 * y + 1;  
        }  
        delay(50);  
    }  
}
```

```
int main() {  
    int gd = DETECT, gm;  
    initgraph(&gd, &gm, "");  
    int xc, yc, r;  
    printf("Enter the center coordinates (xc, yc): ");  
    scanf("%d %d", &xc, &yc);  
    printf("Enter the radius (r): ");  
    scanf("%d", &r);  
    midpointCircle(xc, yc, r);  
    getch();  
    closegraph();    return  
    0;  
}
```

OUTPUT



10. WAP to draw a circle using Bresenham's circle drawing algorithm.

CODE

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

void drawCircle(int xc, int yc, int x, int 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 - x, WHITE);
}

void bresenhamCircle(int xc, int yc, int r) {
    int x = 0, y = r;    int d = 3 - 2 * r;
    drawCircle(xc, yc, x, y);    while (y >= x)
    {        x++;        if (d > 0) {            y--;
d = d + 4 * (x - y) + 10;
        } else {
            d = d + 4 * x + 6;
        }
        drawCircle(xc, yc, x, y);
    }
}
```

Name: Ashish Kothari
Roll No: 2221283 (17)
Subject: Computer Graphics Lab

Course: BCA (VI)
Section: D1
Sub Code: PBC 601

```
}
```

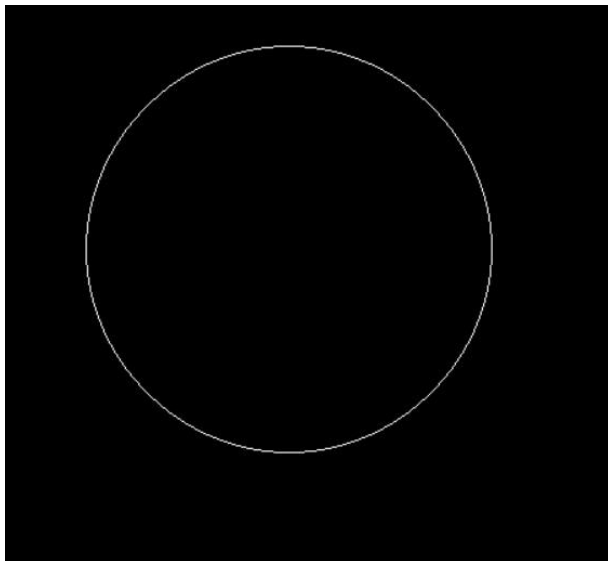
```
int main() {    int gd = DETECT, gm;  
    initgraph(&gd, &gm, "");
```

```
    int xc, yc, r;    printf("Enter center  
coordinates (xc, yc): ");    scanf("%d %d",  
&xc, &yc);    printf("Enter radius: ");  
    scanf("%d", &r);
```

```
    bresenhamCircle(xc, yc, r);
```

```
    getch();  
    closegraph();  
    return 0; }
```

OUTPUT



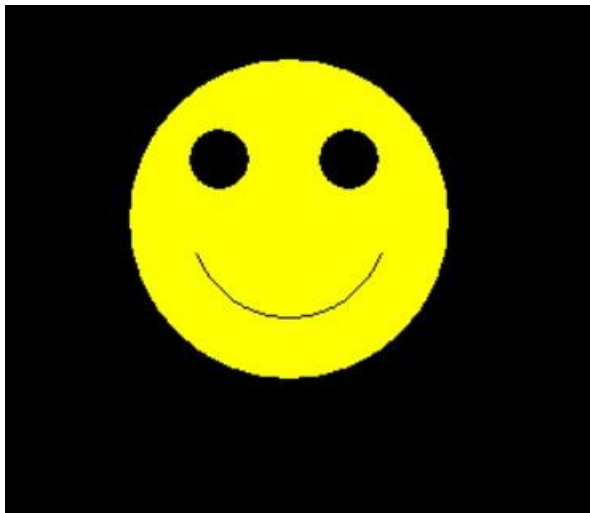
11. WAP to draw a smiley.

CODE

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

int main() {    int gd = DETECT, gm;    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI"); //
Specify the path to BGI folder if needed    setcolor(YELLOW);
setfillstyle(SOLID_FILL, YELLOW);    circle(300, 200, 80);    floodfill(300, 200,
YELLOW);    setcolor(BLACK);    setfillstyle(SOLID_FILL, BLACK);    circle(265,
170, 15);    floodfill(265, 170, BLACK);    circle(330, 170, 15);    floodfill(335, 170,
BLACK);    setcolor(BLACK);    arc(300, 200, 200, 340, 50);    getch();    closegraph();
return 0;
}
```

OUTPUT



12. WAP to draw a hut using filled colors.

CODE

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

int main() {    int gd = DETECT, gm;
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
setcolor(WHITE);    rectangle(100, 200, 300,
300);    line(100, 200, 200, 100);    line(200,
```

Name: Ashish Kothari
Roll No: 2221283 (17)
Subject: Computer Graphics Lab

Course: BCA (VI)
Section: D1
Sub Code: PBC 601

```
100, 300, 200);    rectangle(170, 250, 220, 300);  
setfillstyle(SOLID_FILL, BROWN);  
floodfill(101, 201, WHITE);  
setfillstyle(SOLID_FILL, RED);  
floodfill(200, 101, WHITE);  
setfillstyle(SOLID_FILL, BLUE);  
floodfill(151, 251, WHITE);    getch();  
closegraph();    return 0;  
}
```

OUTPUT

