

Computer Graphics:

Submitted to: Mrs. Simmi Sahani

Submitted By:

Bhupender

Roll no. 2038

BCA 6th semester

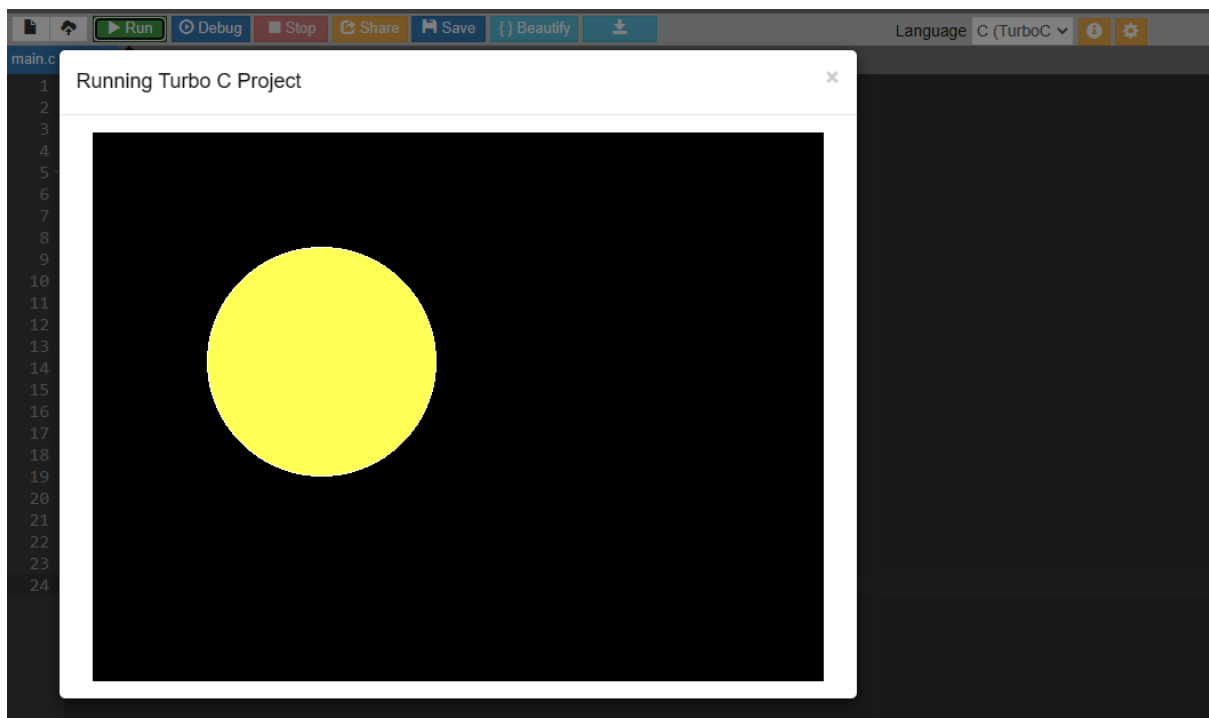
Govt. College Solan

PRACTICAL 1:

Draw a circle. Fill the circle with yellow color

```
main.c
1 #include <stdio.h>
2 #include <conio.h>
3 #include <graphics.h>
4
5 int main() {
6     int gd = DETECT, gm;
7     int x = 200, y = 200, r = 100;
8
9     initgraph(&gd, &gm, "C:\\TC\\BGI"); // Initialize graphics mode
10
11     // Set fill style to solid fill with yellow color
12     setfillstyle(SOLID_FILL, YELLOW);
13
14     // Draw circle with center (x,y) and radius r
15     circle(x, y, r);
16
17     // Fill circle with yellow color using floodfill
18     floodfill(x, y, WHITE);
19
20     getch();
21     closegraph();
22     return 0;
23 }
24
```

Output:

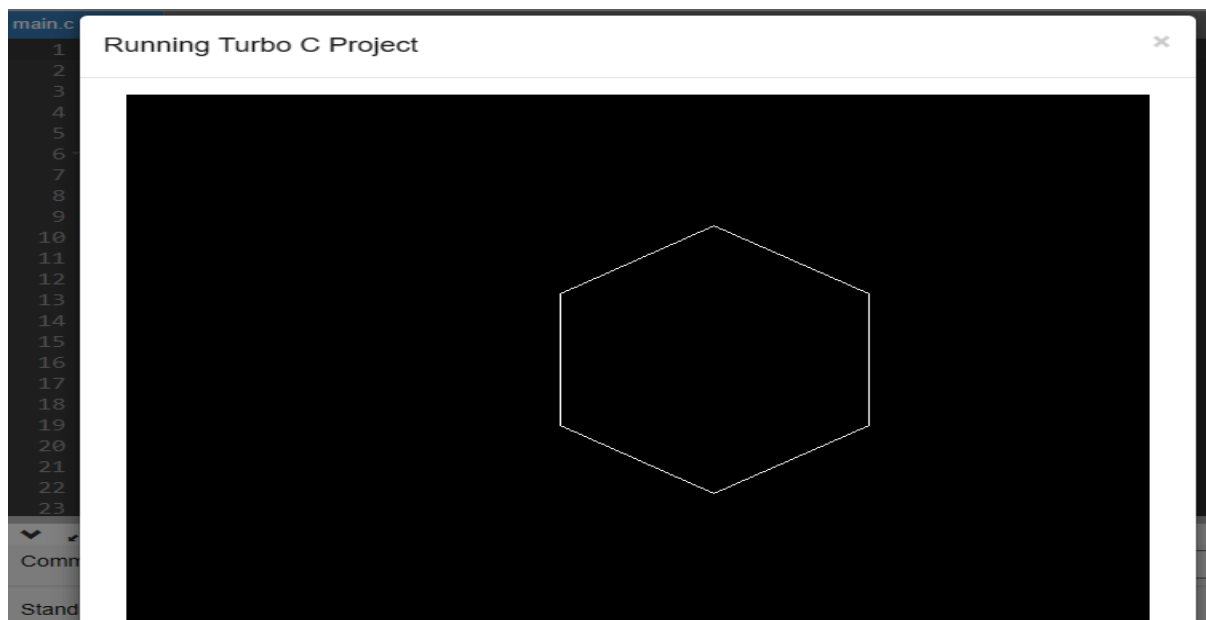


PRACTICAL 2:

Draw Hexagon

```
main.c
1  #include <stdio.h>
2  #include <graphics.h>
3  #include <conio.h>
4
5  int main()
6  {
7      int gd = DETECT, gm;
8      initgraph(&gd, &gm, "C:\\\\TC\\\\BGI");
9
10     // Hexagon
11     line(367, 119, 464, 181);
12     line(464, 181, 464, 301);
13     line(464, 301, 367, 363);
14     line(367, 363, 271, 301);
15     line(271, 301, 271, 181);
16     line(271, 181, 367, 119);
17
18     getch();
19     closegraph();
20     return 0;
21 }
22
```

Output:

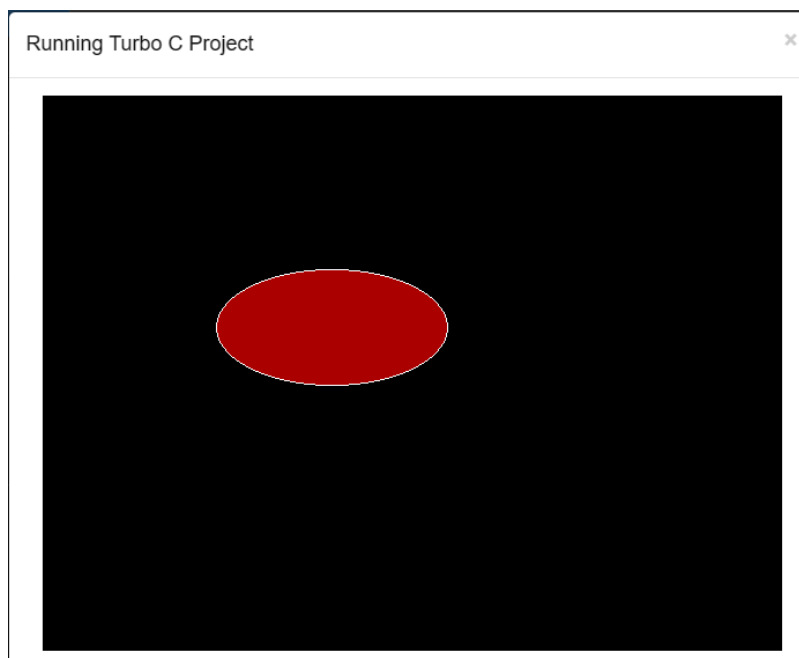


Practical 3

Draw a complete ellipse filled with red color.

```
main.c
1  #include<stdio.h>
2  #include <graphics.h>
3
4  int main()
5  {
6      int gd = DETECT, gm;
7
8      // location of ellipse
9      int x = 250, y = 200;
10     // here is the starting angle and end angle
11     int start_angle = 0;
12     int end_angle = 360;
13     // radius from x axis and y axis
14     int x_rad = 100;
15     int y_rad = 50;
16
17     initgraph(&gd, &gm, "C:\\TC\\BGI");
18
19     setfillstyle(SOLID_FILL, RED);
20     // ellipse function
21     ellipse(x, y, start_angle,
22            end_angle, x_rad, y_rad);
23
24     fillellipse(x, y, x_rad, y_rad);
25
26     getch();
27     closegraph();
28     return 0;
29 }
```

output



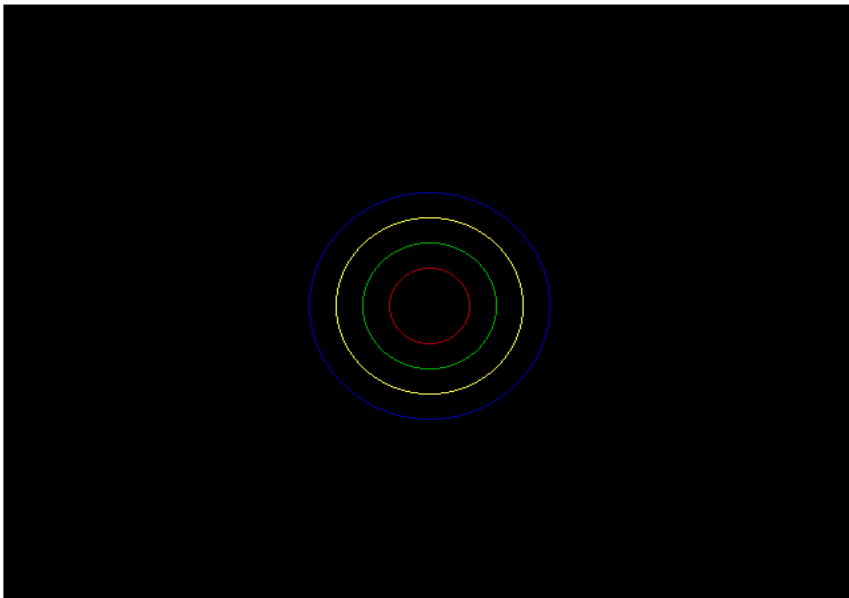
PRACTICAL 4:

Draw concentric circles with different color lines.

```
main.c
1 #include<stdio.h>
2 #include<graphics.h>
3
4 int main(){
5     int gd = DETECT, gm;
6     int x, y;
7     initgraph(&gd, &gm, "C:\\TC\\BGI");
8     /* Initialize center of circle with center of screen */
9     x = getmaxx()/2;
10    y = getmaxy()/2;
11
12    /* Draw circles on screen */
13    setcolor(RED);
14    circle(x, y, 30);
15    setcolor(GREEN);
16    circle(x, y, 50);
17    setcolor(YELLOW);
18    circle(x, y, 70);
19    setcolor(BLUE);
20    circle(x, y, 90);
21
22    getch();
23    closegraph();
24    return 0;
25 }
```


Output:

Running Turbo C Project



PRACTICAL 5:

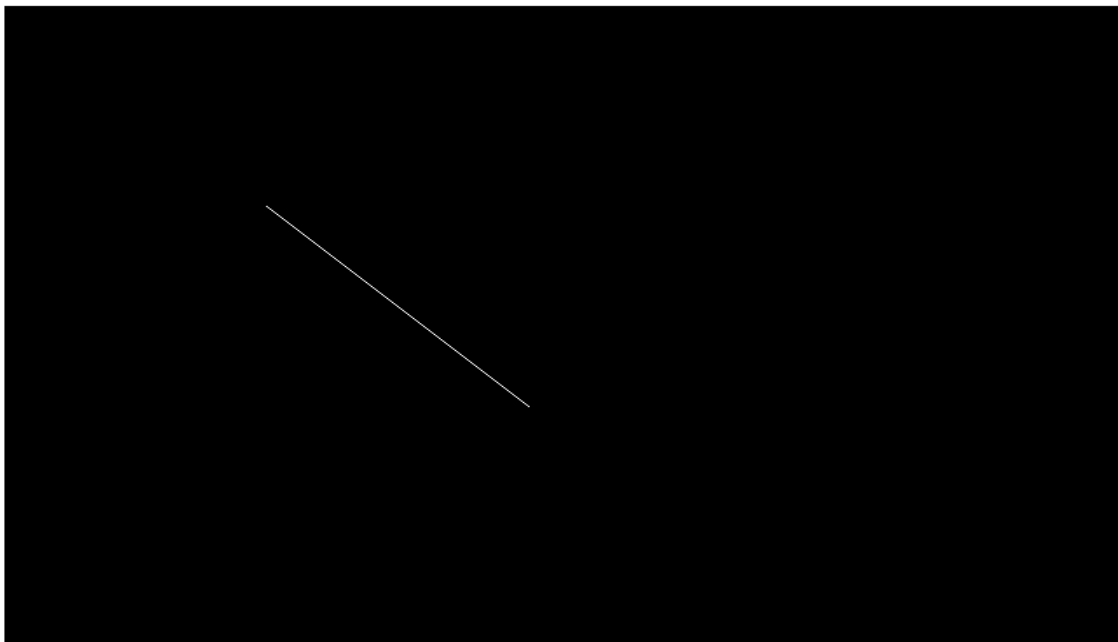
Draw a line where coordinates are (150, 150, 300, 300).



```
1 #include <stdio.h>
2 #include <graphics.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int gd = DETECT, gm;
8
9     initgraph(&gd, &gm, "C:\\TC\\BGI");
10
11     line(150, 150, 300, 300);
12
13     getch();
14     closegraph();
15     return 0;
16 }
```

Output:

Running Turbo C Project

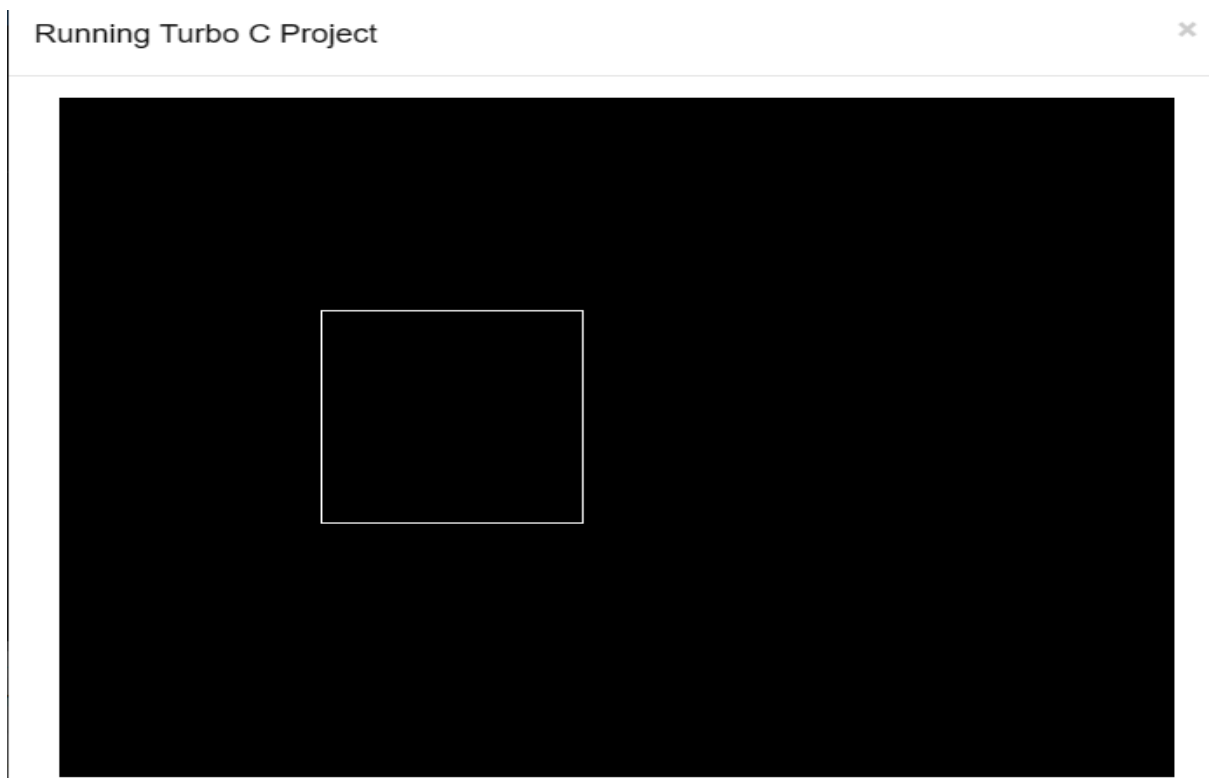


PRACTICAL 6:

Draw a rectangle with the coordinates of (150, 150, 300, 300).

```
main.c
1 #include <stdio.h>
2 #include <graphics.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int gd = DETECT, gm;
8
9     initgraph(&gd, &gm, "C:\\TC\\BGI");
10
11     rectangle(150, 150, 300, 300);
12
13     getch();
14     closegraph();
15     return 0;
16 }
17
18
19
```

Output:



PRACTICAL 7:

Draw a hut.

```
main.c
1 #include <stdio.h>
2 #include <graphics.h>
3 #include <conio.h>
4 int main()
5 {
6     int gd = DETECT, gm;
7     initgraph(&gd, &gm, "C:\\TC\\BGI");
8     /* Draw Hut */
9     setcolor(WHITE);
10    rectangle(150,180,250,300);
11    rectangle(250,180,420,300);
12    rectangle(180,250,220,300);
13    line(200,100,150,180);
14    line(200,100,250,180);
15    line(200,100,370,100);
16    line(370,100,420,180);
17    /* Fill colours */
18    setfillstyle(SOLID_FILL, BROWN);
19    floodfill(152, 182, WHITE);
20    floodfill(252, 182, WHITE);
21    setfillstyle(SLASH_FILL, BLUE);
22    floodfill(182, 252, WHITE);
23    setfillstyle(SOLID_FILL, BROWN);
24    floodfill(200, 105, WHITE);
25    floodfill(210, 105, WHITE);
26
27    getch();
28    closegraph();
29    return 0;
30 }
```

Output:

