

## Code:

```
1  #include <stdio.h>
2  #include <conio.h>
3  #include <graphics.h>
4  void BoundaryFill(int x, int y, int fill, int boundary);
5  void FloodFill(int x, int y, int floodFill, int interiorColor);
6  int main()
7  {
8      int gd = DETECT, gm, length, choice;
9      initgraph(&gd, &gm, "C:\\\\TURBOC3\\\\BGI");
10     printf("Enter the length of the square: ");
11     scanf("%d", &length);
12     printf("Choose the Algorithm to use:\n1. Boundary Fill\n2. Flood Fill");
13     scanf("%d", &choice);
14     rectangle((getmaxx() - length) / 2, (getmaxy() - length) / 2, (getmaxx() +
↪ length) / 2, (getmaxy() + length) / 2);
15     switch (choice)
16     {
17     case 1:
18         BoundaryFill(getmaxx() / 2, getmaxy() / 2, 12, WHITE);
19         break;
20
21     case 2:
22         FloodFill(getmaxx() / 2, getmaxy() / 2, 11, BLACK);
23         break;
24     default:
25         break;
26     }
27     getch();
28     closegraph();
29     return 0;
30 }
31 void BoundaryFill(int x, int y, int fill, int boundary)
32 {
33     int current;
34     current = getpixel(x, y);
35     if(current != fill && current != boundary)
36     {
37         putpixel(x, y, fill);
38         BoundaryFill(x + 1, y, fill, boundary);
39         BoundaryFill(x, y + 1, fill, boundary);
40         BoundaryFill(x - 1, y, fill, boundary);
41         BoundaryFill(x, y - 1, fill, boundary);
42     }
43 }
44 void FloodFill(int x, int y, int floodFill, int interiorColor)
```

```
45 {
46     unsigned int current;
47     current = getpixel(x, y);
48     if(current == interiorColor)
49     {
50         putpixel(x, y, floodFill);
51         FloodFill(x + 1, y, floodFill, interiorColor);
52         FloodFill(x, y + 1, floodFill, interiorColor);
53         FloodFill(x - 1, y, floodFill, interiorColor);
54         FloodFill(x, y - 1, floodFill, interiorColor);
55     }
56 }
```