Minesweeper: Write a program that takes 3 arguments from user M, N, and p and produces an M–by–N boolean array where each entry is occupied with probability p. In the minesweeper game, occupied cells represent bombs and empty cells represent safe cells. Print out the array using an asterisk for bombs and a period for safe cells. Then, replace each safe square with the number of neighboring bombs (above, below, left, right, or diagonal) and print out the solution.

```
* * . . .        * * 1 0 0
. . . . .        3 3 2 0 0
. * . . .        1 * 1 0 0
```

Try to write your code so that you have as few special cases as possible to deal with, by using an (M+2)–by–(N+2) boolean array.

Code:-

```java
public class Minesweeper {
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.print("Please Enter M: ");
        int m = sc.nextInt();
        System.out.print("Please Enter N: ");
        int n = sc.nextInt();
        System.out.print("Please Enter P: ");
        double p = sc.nextDouble();

        boolean[][] matrix = new boolean[m + 2][n + 2];
        for (int i = 1; i <= m; i++) {
            for (int j = 1; j <= n; j++) {
                matrix[i][j] = (Math.random() < p);
                if (matrix[i][j]) System.out.print("* ");
                else System.out.print("- ");
            }
            System.out.println();
        }
```

```java
        int[][] res = new int[m + 2][n + 2];
        int[] dx = {0, 0, 1, -1, -1, 1, -1, 1};
        int[] dy = {1, -1, 0, 0, -1, 1, 1, -1};
        for (int i = 1; i <= m; i++) {
            for (int j = 1; j <= n; j++) {
                for (int k = 0; k < 8; k++)
                    if (matrix[i + dx[k]][j + dy[k]])
                        res[i][j]++;
            }
        }
        System.out.println("----------------------------------");
        for (int i = 1; i <= m; i++) {
            for (int j = 1; j <= n; j++) {
                if (matrix[i][j])
                    System.out.print("* ");
                else
                    System.out.print(res[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Output:-