1. The complexity is N2 because once we iterate from 0 to arr.length(the for loop) and then everytime the while loop, inside the for loop, is performed exactly n times because of its implementation.
2. In the best case the running time will be N when the elements in the matrix are odd numbers. In the worst case the running time will be N \* M when all the elements are even numbers. I’m not quite sure, but as a prediction, in my opinion, the average case will be N \* log(M).
3. When the variable row is 1, the running time is N \* M (printing the whole matrix elements’) which is the worst case. The best case (M) is when row is equal to the array.length – 1. The exact work time is ((row \* N) / N) \* M. The average is (N / 2) \* M. The work time, as we obviously find out, depends on the row value. As the row value increases, the work time decreases.