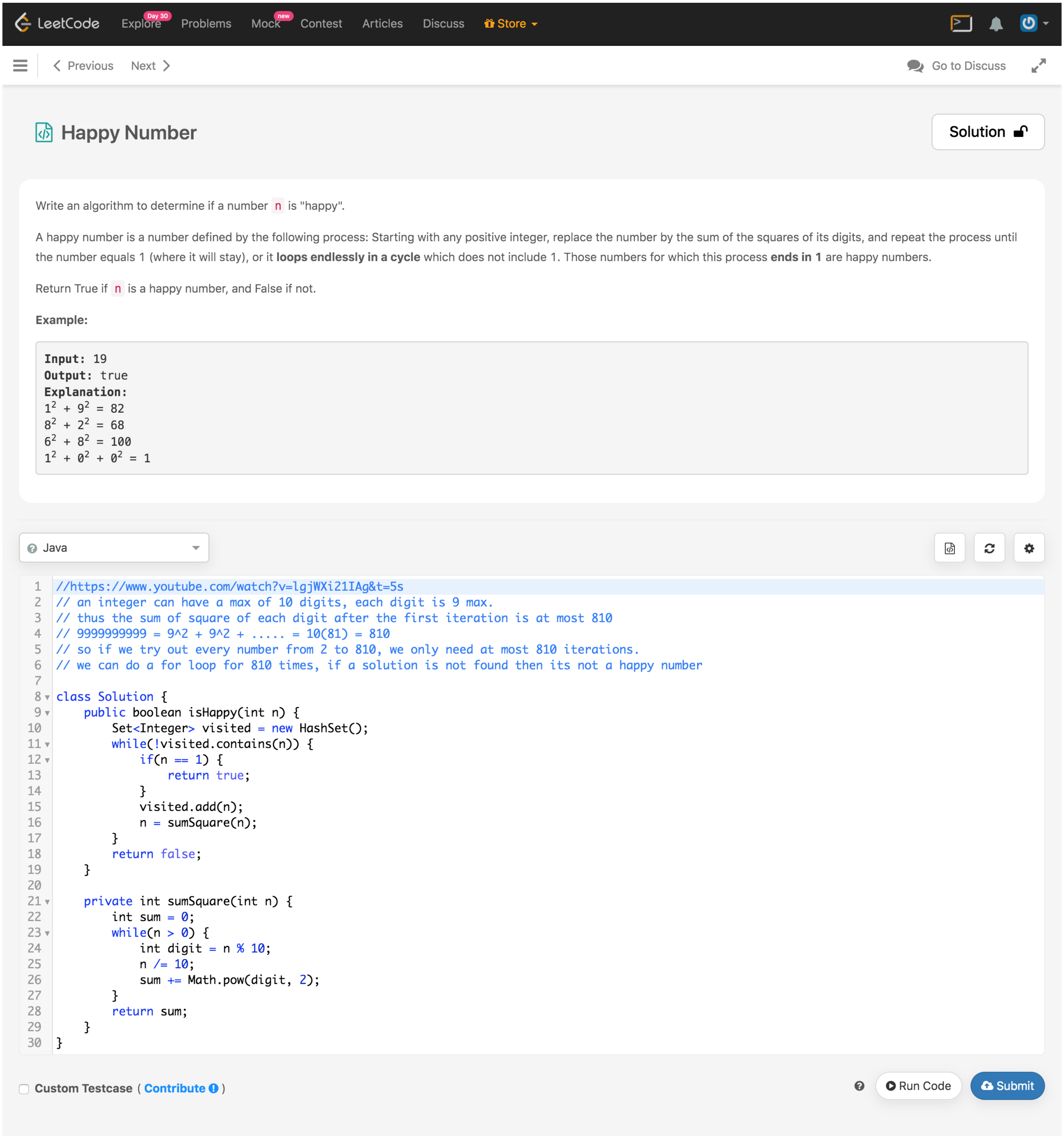

Happy Number

Write an algorithm to determine if a number `n` is "happy".

A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it **loops endlessly in a cycle** which does not include 1. Those numbers for which this process **ends in 1** are happy numbers.

Return True if `n` is a happy number, and False if not.

Example:

Input: 19
Output: true
Explanation:

$1^2 + 9^2 = 82$

$8^2 + 2^2 = 68$

$6^2 + 8^2 = 100$

$1^2 + 0^2 + 0^2 = 1$

```java
//https://www.youtube.com/watch?v=lgjWXi21IAg&t=5s
// an integer can have a max of 10 digits, each digit is 9 max.
// thus the sum of square of each digit after the first iteration is at most 810
// 9999999999 = 9^2 + 9^2 + ..... = 10(81) = 810
// so if we try out every number from 2 to 810, we only need at most 810 iterations.
// we can do a for loop for 810 times, if a solution is not found then its not a happy number

class Solution {
    public boolean isHappy(int n) {
        Set<Integer> visited = new HashSet();
        while(!visited.contains(n)) {
            if(n == 1) {
                return true;
            }
            visited.add(n);
            n = sumSquare(n);
        }
        return false;
    }

    private int sumSquare(int n) {
        int sum = 0;
        while(n > 0) {
            int digit = n % 10;
            n /= 10;
            sum += Math.pow(digit, 2);
        }
        return sum;
    }
}
```