Practice 02:

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.
* 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.

Return true *if* n *is a happy number, and* false *if not*.

**Example 1:**

**Input:** n = 19

**Output:** true

**Explanation:**

12 + 92 = 82

82 + 22 = 68

62 + 82 = 100

12 + 02 + 02 = 1

**Example 2:**

**Input:** n = 2

**Output:** false

**Constraints:**

* 1 <= n <= 231 - 1

Code:

class Solution {

private int getNext(int n) {

int totalSum = 0;

while (n > 0) {

int d = n % 10;

n = n / 10;

totalSum += d \* d;

}

return totalSum;

}

public boolean isHappy(int n) {

Set<Integer> seen = new HashSet<>();

while (n != 1 && !seen.contains(n)) {

seen.add(n);

n = getNext(n);

}

return n == 1;

}

}