CS 261 Recitation 8: Maps

In order to get credit for the recitation, you need to be checked off by the end of recitation. For non-zero recitations, you can earn a maximum of 3 points for recitation work completed outside of recitation time, but you must finish this recitation before the next recitation. For extenuating circumstance, contact your recitation TAs and Instructor.

Group work, and individual work are highlighted

Recitation 9 Grade Breakdown:

• Part 1: Find the missing number	5 pts
• Part 2: Determine happy number	5 pts

Download and unzip the start code: (wget command is recommended)

wget https://classes.engr.oregonstate.edu/eecs/summer2022/cs261-001/recitations/rec8.zip

Part 1: Find the missing number

Problem statement: Given an array `nums` containing `n` distinct numbers in the range `[0, n]`, **use maps** to return the only number in the range that is missing from the array.

Example 1:

```
Input: nums = [3,0,1]

Output: 2

Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3].
2 is the missing number in the range since it does not appear in nums.
```

Example 2:

```
Input: nums = [0,1]

Output: 2

Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2].
2 is the missing number in the range since it does not appear in nums.
```

Example 3:

```
Input: nums = [9,6,4,2,3,5,7,0,1]

Output: 8

Explanation: n = 9 since there are 9 numbers, so all numbers are in the range [0,9].
8 is the missing number in the range since it does not appear in nums.
```

Constraints:

- `n == nums.length`
- 1 <= `n` <= 10^4
- 0 <= `nums[i]` <= n

Part 2: Determine happy number

Problem Statement: Write a function that uses maps 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:
```

$1^2 + 9^2 = 82$

$8^2 + 2^2 = 68$

$6^2 + 8^2 = 100$

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

Example 2:

```
Input: n = 2

Output: false
```

Constraints:

- $1 <= n <= 2^{31} - 1$

<u>Make sure you get checked off</u> by showing the output of your program, your report, and your group work before the end of your recitation section.

For backup purposes, please submit your work for this recitation (including all documents/text files for group work, and programs) to TEACH.