

# Happy Number

Try to solve the Happy Number problem.

## We'll cover the following

- Statement
- Examples
- Test your understanding of the problem
- Figure it out
- Try it yourself

## Statement

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

We use the following process to check if a given number is a happy number:

- Starting with the given number  $n$ , replace the number with the sum of the squares of its digits.
- Repeat the process until:
  - The number equals 1, which will depict that the given number  $n$  is a happy number.
  - It enters a cycle, which will depict that the given number  $n$  is not a happy number.

Return TRUE if  $n$  is a happy number, and FALSE if not.

## Constraints

- $1 \leq n \leq 2^{31} - 1$

## Examples

### Sample example 1

#### Inputs

$n$	23
-----	----

#### Explanation

$$2^2 + 3^2 = 13$$

#### Output

?

We added the square of the two digits, and got the value of  $n$  for the next iteration.

1 of 6



## Test your understanding of the problem



Let's take a moment to make sure we have correctly understood the problem. The quiz below helps us to check that we are solving precisely the right problem:

Happy Number

1

Is 28 a Happy Number?

A) TRUE

B) FALSE

Submit Answer

<

Question 1 of 20 attempted

>

Reset Quiz ↻

Figure it out

We have a game for you to play: re-arrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to re-arrange them in the correct sequence.

Initialise a variable `slow` with the input number and `fast` with the squared sum of the input number's digits.

If `fast` is not 1 and also not equal to `slow`, increment `slow` by one iteration and `fast` by two iterations. Essentially, set `slow` to `Sum of Digits(slow)` and `fast` to `Sum of Digits(Sum of Digits(fast))`.

If `fast` converges to 1, return TRUE, otherwise return FALSE.

Reset

Show Solution

Submit

?

Tt

🔄

# Try it yourself



template in the other file that you may build on to solve this problem.

main.java

SumOfSquaredDigits.java

```
1 import java.util.*;
2 public class Main{
3     public static boolean isHappyNumber(int n) {
4
5         // Write your code here
6
7         return true;
8     }
9 }
```

Powered by AI

▶

Submit

Test Cases

Results

Case 1

Case 2

Case 3

Input #1

2147483646

Happy Number

← Back

Fast and Slow Pointer...

Next →

Solution: Happy Num...

☒ Mark as Completed