

In mathematics, the **Champernowne** constant is a **transcendental real constant** whose decimal expansion has important properties.

For base 10, the number is defined by concatenating representations of successive integers:  
 $C_{10} = 0.12345678910111213141516...$

Given a string, output the position of its first occurrence in the fractional part of Champernowne constant.

### Example

- For `pattern = "123"`, the output should be `champernowne(123) = 1`.  
"123" first appears at the 1st decimal.
- For `pattern = "56"`, the output should be `champernowne(56) = 5`.  
"56" first appears at the 5th decimal.
- For `pattern = "213"`, the output should be `champernowne(213) = 14`.  
"213" first appears at the 14th decimal.

### Input/Output

- [execution time limit] 4 seconds (js)**
- [input] string pattern**

*Guaranteed constraints:*

$1 \leq \text{pattern.length} \leq 1000$ .

- [output] integer64**

*Guaranteed constraints:*

$1 \leq \text{output} < 2^{53}$

*As implied by the constraint above, the pattern must be present somewhere between those bounds.*

### [JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
  console.log("This prints to the console when you Run Tests");
  return "Hello, " + name;
```

