DMPG '18 S1 - Mimi and Types

Time limit: 2.0s **Memory limit:** 128M

Mimi was writing code in C++. She was utilizing a `std::tuple` for some purpose. Because her definition of the data structure was very complicated, she did not want to type it every single time. So therefore she decided to use numbers to represent the type instead.

A **type** is valid if and only if it consists of

- a single `0` only.
- a single digit n $(1 \le n \le 9)$ followed by exactly n valid **types**.

But soon, things got out of hand. Mimi wasn't sure whether some of her types were valid. Could you help her validate her types?

Constraints

For all subtasks:

S will be non-empty and will be at most 10^5 characters in length.

All characters in S will be digits.

Subtask 1 [5%]

S will only contain `0`.

Subtask 2 [15%]

S will only contain `0` and `1`.

Subtask 3 [80%]

No additional constraints.

Input Specification

The first and only line of input will contain a non-empty string S consisting only of digits, representing the type.

Output Specification

The output should contain `Valid` if the type is valid, and `Invalid` if the type is invalid.

Sample Input 1

```
2300040000
```

Sample Output 1

```
Valid
```

Sample Input 2

```
23000
```

Sample Output 2

```
Invalid
```

Sample Input 3

```
00
```

Sample Output 3

```
Invalid
```