

# Verifying Validity (100 points)

## Introduction

Vanessa manages an online book catalog and has to manually input any new books into the system. Since she sometimes mistypes the ISBN, she would like a program to check if a given ISBN is valid.

To check the validity of an ISBNs a checksum is calculated from its digits specifically for the purpose of catching transcription errors. The checksum is calculated as the sum of all digits, each multiplied by its (integer) weight. Weights are assigned based on the position of the digit in the ISBN and alternate between 1 and 3. For example the first digit will get a weight of 1, the second will get a weight of 3, the third will get 1 again and so on.

To be a valid ISBN, the checksum must be a multiple of 10.

For example, 1234567890128 would be a valid ISBN, because  $1*1 + 2*3 + 3*1 + 4*3 + 5*1 + 6*3 + 7*1 + 8*3 + 9*1 + 0*3 + 1*1 + 2*3 + 8*1 = 100$  which is divisible by 10.

If the last digit was 3 instead of 8 i.e. 1234567890123, using the formula we would only get 95, which is not divisible by 10, thus this would be an invalid ISBN.

## Input Specifications

Your program must read from **STDIN**:-

A single integer  $N$  ( $1 \leq N \leq 10$ ) indicating the number of ISBN entries to verify.

$N$  lines each containing a single ISBN (ISBNs are 13-digit integers, between 1000000000000 and 9999999999999 inclusive).

## Output Specifications

Based on the input, print out either "VALID" or "NOT VALID" for each ISBN.

## Sample Input/Output

### Input

```
4
1234567890123
1234567890128
1111111111116
1111111111118
```

### Output

```
NOT VALID
VALID
VALID
NOT VALID
```

## Explanation

By applying the formula used above, we check if the ISBN is valid.