Problem 192: By the Book

Difficulty: Easy

Author: Brett Reynolds, Annapolis Junction, Maryland, United States

Originally Published: Code Quest Community College Outreach 2022

Problem Background

When searching for books in the library, it can sometimes be difficult to be certain if you've gotten the correct book. Particularly with reference books or other textbooks, titles can be the same or similar and it's not always easy to tell who the author is. Fortunately, publishers have developed a way to be certain you've got not only the correct book, but the correct version of it: the International Standard Book Number, or ISBN.

An ISBN is a 10- or 13- digit code (books published since 2007 have 13-digit codes) that is supposed to uniquely identify a particular edition of a book (although in practice, there are some overlaps). Since accurately remembering or writing down a long string of numbers is not guaranteed, the ISBN contains a built-in error-checking mechanism. The last digit in an ISBN is known as a "check digit" and can be used to verify that all of the preceding digits represent a valid ISBN.

Problem Description

Some Lockheed Martin sites maintain a library of technical books for employees to check out; this helps employees learn about technologies they may not be familiar with. Unfortunately, your site suffered a fire recently, and you've volunteered to help assess the damage and try to recover the collection. The library has a collection of books (all published before 2007) with labels that were damaged in the fire. Someone has already attempted to restore the ISBN numbers on these labels, but it quickly became apparent that not all of the numbers are correct. You have been asked to write a program that can read in ISBN numbers from these labels and determine if the numbers are valid or invalid. Books with invalid ISBNs will be set aside to be manually researched and corrected.

To calculate the check digit of an ISBN-10, each of the first nine digits is multiplied by an "integer weight," a number calculated by subtracting the index of the digit from 10. These products are then added together to form a single number. The check digit is the number that must be added to that sum to reach a multiple of 11 (if that number is 10, the check digit is the letter 'X').

For example, consider the ISBN-10 number 0-306-40615-2:

$$S = \sum_{i=0}^{8} x_i \times (10 - i)\,;\;\; x = \{0,3,0,6,4,0,6,1,5,2\}$$

$$\begin{aligned} S = &\big(0 \times (10-0)\big) + \big(3 \times (10-1)\big) + \big(0 \times (10-2)\big) + \big(6 \times (10-3)\big) + \big(4 \times (10-4)\big) \\ &+ \big(0 \times (10-5)\big) + \big(6 \times (10-6)\big) + \big(1 \times (10-7)\big) + \big(5 \times (10-8)\big) \end{aligned}$$

$$S = (0 \times 10) + (3 \times 9) + (0 \times 8) + (6 \times 7) + (4 \times 6) + (0 \times 5) + (6 \times 4) + (1 \times 3) + (5 \times 2)$$
$$S = 0 + 27 + 0 + 42 + 24 + 0 + 24 + 3 + 10$$
$$S = 130$$

The weighted sum of this ISBN is therefore 130. 130 is not divisible by 11; the next highest multiple of 11 is 132 (11 * 12 = 132). Since 132 - 130 is 2, the check digit for this ISBN should be 2. This matches the last digit in the ISBN we were investigating, so this ISBN number appears to be valid.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include a single line containing an ISBN-10 number to be checked. ISBNs can include numeric digits and the uppercase letter X.

```
3
0306406152
0306401652
080442957X
```

Sample Output

For each test case, your program must print, on a single line, the word VALID if the given ISBN number has a valid check digit, or the word INVALID if it does not.

```
VALID
INVALID
VALID
```