3. Credit Card

Program Name: CreditCard.java Input File: creditcard.dat

A Visa or MasterCard has a 16-digit credit card number. The issuer identifier number is denoted by the first 6 digits. The last digit is known as the check digit and is generated to satisfy the Luhn check. If you take away the 6 identifying digits and the last check digit, then you have the 9 digits that form your account number.

The following algorithm gives the Luhn check:

1. Starting with the leftmost digit, double every second digit (going from left to right). Do **not** double the check digit. In all you will double 8 digits.
2. If the doubling of a digit results in a 2-digit number, add the two digits to form a single digit.
3. Replace the corresponding digits in the original credit card number with the new digits. There will be 8 replacements in alternate positions.
4. Add all the 16 digits in the new set of numbers.
5. If the sum is evenly divisible by 10 then it is a valid credit card number, otherwise it is not.

Let us suppose that the credit card number is 5490 1234 5678 9128, as shown in the first row below with the digits separated.

5	4	9	0	1	2	3	4	5	6	7	8	9	1	2	8
10		18		2		6		10		14		18		4	
1		9		2		6		1		5		9		4	
1	4	9	0	2	2	6	4	1	6	5	8	9	1	4	8

The second row above shows every second digit doubled, starting from the left. The third row shows the doubled digits from row two, with the 2-digit numbers converted to single digits by adding the digits together. The last row shows the number with the changed digits from row three substituted for the original digits from row one.

If you add the last row of 16 digits you get 70 which is evenly divisible by 10, and hence 5490 1234 5678 9128 is a valid credit card number.

Input

The first line will contain a single integer `n` that indicates the number of credit card numbers to test. The next `n` lines will each contain a single 16-digit credit card number (without any spaces).

Output

For each card, print `VALID` if the card is a valid card or print `INVALID` if the card is not a valid card.

Example Input File

```
4
1234567891234567
4552720412345677
9087654321135793
5490123456789128
```

Example Output to Screen

```
INVALID
VALID
INVALID
VALID
```