

## 3228 Secret Sharing

In general, many cryptosystems use one secret key to access encrypted files. If an attacker hacks in and steals the secret key, the information can be leaked out. *Secret sharing* is a technique to distribute a secret key among a group of participants. A secret sharing scheme divides a secret key into  $N$  pieces, which are called *shares*. Anyone with fewer than  $N$  shares cannot regenerate the secret key and access the secret information. For example, consider the secret key “password” is divided into four shares “pa”, “ss”, “wo”, and “rd”. All of which are required to recover the original secret key. However to recover the key, you need not only the shares but also the order of shares. If the order is not acquired, you could produce a wrong key, such as “wosspard” or “pawordss”.

Recently, a new cryptographic algorithm, which uses a secret sharing technique to generate a secret key, is developed in your research group. The algorithm uses a very long decimal number as a secret key. The key starts with a non-zero digit. The key is divided into multiple shares. To settle the ordering problem from the shares, a simple rule is adopted; the secret key is the smallest number that can be obtained from the shares. For instance, suppose that a secret key is divided into five shares “2”, “4”, “11”, “33”, and “00”. “24113300” and “11233400” are not the smallest numbers. “00112334” is invalid since the secret key can not start with “0”. The satisfying key is “11002334”.

Write a program that finds the secret key from the shares.

### Input

The input consists of  $T$  test cases. The number of test cases  $T$  is given in the first line of the input file. Each test case starts with a line containing an integer  $N$ , the number of shares,  $1 \leq N \leq 50$ . On next line,  $N$  shares are given, where each share consists of at most 5 digits.

### Output

Print exactly one line for each test case. For each test case, print a secret key. If you can not find a key that satisfies the condition, print ‘INVALID’.

### Sample Input

```
4
5
2 4 11 33 00
3
20 202 2020
6
3 4 5 3 44 555
3
0 00 007
```

### Sample Output

```
11002334
202020202
334445555
INVALID
```