

**AS** (Association for Strings) is an organization which represents strings in different manner. They used to provide different rules concerning string representation. Their service is needed in a country named Bangladesh. Elora, the current chief of the country ask for a rule to AS. AS provides the rule and now Elora is worried about whether the rule is valid or not, if it is valid she also wants to know the lexicographically smallest string which can be represented by the rule. So, she wants to hire a programmer who will do the job for her.

A suffix of a string  $X$  is a string obtained by removing **zero or more** character(s) from the **beginning** of that string. A prefix of a string  $X$  is a string obtained by removing **zero or more** character(s) from the **end** of  $X$ . A proper prefix/suffix of  $X$  is a prefix/suffix with length strictly less than  $X$ .

AS provided the rule as an array  $P$  of length  $N$ . A string  $S$  of length  $N$  can be represented by  $P$ .

### The rule is:

$P[k]$  = length of the longest prefix of  $S[1 \dots k]$  which is proper suffix of  $S[1 \dots k]$ .  $S[1 \dots k]$  denotes the  $k$ -length prefix of  $S$ .

Strings are composed of lowercase letters from 'a' to 'q'.

### Example:

$N = 5$ ,  $P = \{0, 0, 1, 2, 3\}$ ,  $S = \text{"ababa"}$

$P[4]$  is 2 because "ab" is the longest prefix of  $S[1 \dots 4]$  which is also a suffix of it.

You have to check if any string  $S$  can be generated from  $P$ . If it is possible to generate a string  $S$  from  $P$  then you have to output lexicographically smallest  $S$  otherwise output '-1' (quotes for clarity). Please read sample input and output for more clarification.

## Input

The first line contains number of test cases,  $T$ . ( $1 \leq T \leq 15$ ). For each test case the first line contains  $N$  ( $1 \leq N \leq 100000$ ) which is the length of  $P$  and the second line contains  $N$  integers separated by space where the  $i$ -th integer is  $P[i]$ .

## Output

For each case, print a line of the form 'Case  $< x >$ :  $< y >$ ', where  $x$  is the case number and  $y$  is the answer.

## Sample Input

```
3
6
0 0 1 2 0 0
6
1 2 3 4 5 6
6
0 1 2 3 3 4
```

## Sample Output

```
Case 1: ababbb
Case 2: -1
Case 3: -1
```