

Permutations of Strings



Strings are usually ordered in lexicographical order. That means they are ordered by comparing their leftmost different characters. For example, $abc < abd$ because $c < d$. Also $z > yyy$ because $z > y$. If one string is an exact prefix of the other it is lexicographically smaller, e.g., $gh < ghij$.

Given an array of strings sorted in lexicographical order, print all of its permutations in strict lexicographical order. If two permutations look the same, only print one of them. See the 'note' below for an example.

Complete the function `next_permutation` which generates the permutations in the described order.

For example, $s = [ab, bc, cd]$. The six permutations in correct order are:

```
ab bc cd
ab cd bc
bc ab cd
bc cd ab
cd ab bc
cd bc ab
```

Note: There may be two or more of the same string as elements of s .

For example, $s = [ab, ab, bc]$. Only one instance of a permutation where all elements match should be printed. In other words, if $s[0] == s[1]$, then print either $s[0] s[1]$ or $s[1] s[0]$ but not both.

A three element array having three discrete elements has six permutations as shown above. In this case, there are three matching pairs of permutations where $s[0] = ab$ and $s[1] = ab$ are switched. We only print the three visibly unique permutations:

```
ab ab bc
ab bc ab
bc ab ab
```

Input Format

The first line of each test file contains a single integer n , the length of the string array s .

Each of the next n lines contains a string $s[i]$.

Constraints

- $2 \leq n \leq 9$
- $1 \leq |s[i]| \leq 10$
- $s[i]$ contains only lowercase English letters.

Output Format

Print each permutation as a list of space-separated strings on a single line.

Sample Input 0

```
2
ab
cd
```

Sample Output 0

```
ab cd
cd ab
```

Sample Input 1

```
3
a
bc
bc
```

Sample Output 1

```
a bc bc
bc a bc
bc bc a
```

Explanation 1

This is similar to the **note** above. Only three of the six permutations are printed to avoid redundancy in output.