

DAILY ONLINE ACTIVITIES SUMMARY

Date:	12/06/2020	Name:	Raghavendra s
Sem & Sec	8 sem B sec	USN:	4AL16CS071
Online Test Summary			
Subject	BDA		
Max. Marks	60	Score	Not taken due to network problem
Certification Course Summary			
Course	Introduction to R		
Certificate Provider	greatlearning	Duration	3.00hrs
Coding Challenges			
Problem Statement:			
Status: Solved			
Uploaded the report in Github		Uploaded	
If yes Repository name		Raghavendra s	
Uploaded the report in slack		yes	

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

online certificate

greatlearning

Learning for Life

Home

Live Sessions

Certificates

My Courses

</

ONLINE CODDING

HackerRank

PRACTICE

CERTIFICATION

NEW

COMPETE

JOBS

LEADERBOARD

Search

1

raghuyadavs153

Practice > C > Functions > Permutations of Strings

35 more points to get your next star!

Rank: 51510 | Points: 165/200

C

Problem

Submissions

Leaderboard

Discussions

Editorial

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:

Author

Difficulty

Max Score

Submitted By

NEED HELP?

RATE THIS CHALLENGE

MORE DETAILS

Download problem statement

Download sample test cases

Suggest Edits

Facebook

Twitter

LinkedIn

bayleef

Medium

40

13987

View discussions

View editorial

View top submissions

☆☆☆☆☆

Download problem statement

Download sample test cases

Suggest Edits

Facebook

Twitter

LinkedIn

```
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

Program:

```
int next_permutation(int n, char **s){
    // Find non-increasing suffix
    int i = n-1;
    while(i>0 && strcmp(s[i-1],s[i])>=0)
        i--;    // find key
    if (i<=0) return 0;

    // Swap key with its successor in suffix
    int j = n-1;
    while(strcmp(s[i-1],s[j])>=0)
        j--;    // find rightmost successor to key
    char *tmp = s[i-1];
    s[i-1] = s[j];
    s[j] = tmp;
```

```
// Reverse the suffix
j = n-1;
while(i<j) {
    tmp = s[i];
    s[i] = s[j];
    s[j] = tmp;
    i++;
    j--;
}
return 1;
}
```