Permutations

Try to solve the Permutations problem.

We'll cover the following

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

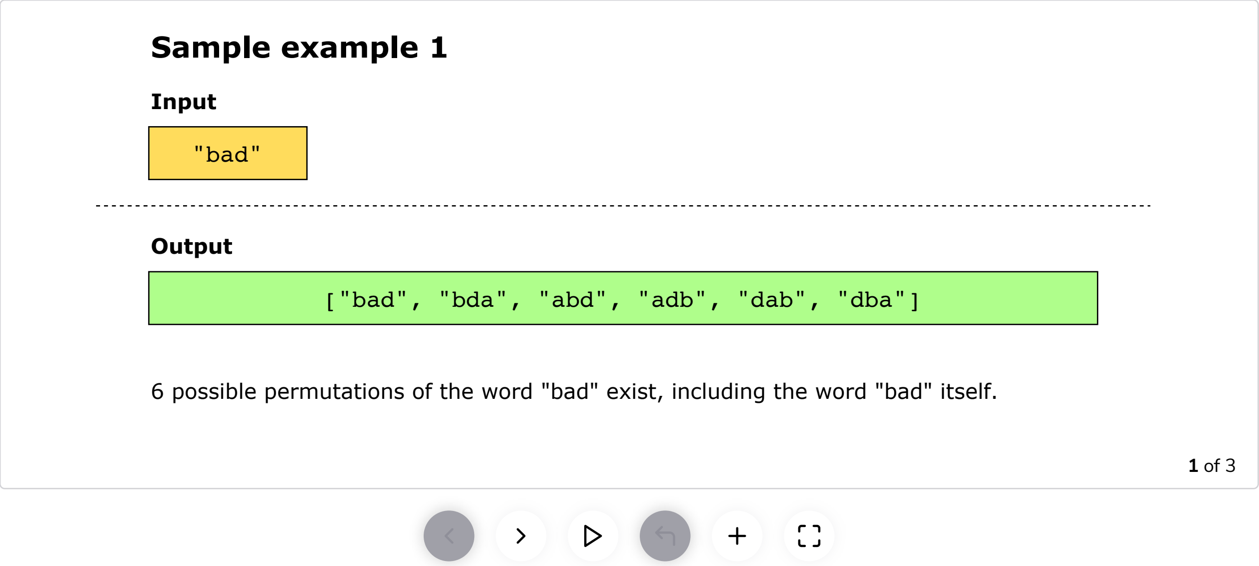
Given an input string, return all possible permutations of the string.

> **Note:** The order of permutations does not matter.

Constraints:

- All characters in the input string are unique.
- $1 \leq$ `word.length` ≤ 6

Examples



Sample example 1

Input

```
"bad"
```

Output

```
["bad", "bda", "abd", "adb", "dab", "dba"]
```

6 possible permutations of the word "bad" exist, including the word "bad" itself.

1 of 3

Understand the problem

Let’s take a moment to make sure you’ve correctly understood the problem. The quiz below helps you check if you’re solving the correct problem:

Permutations

1 What should be the output if the following string is given as input?

Input string = “xyz”

A) [“xyz”, “xzy”, “yxz”, “yzx”, “zyx”, “zxy”]

B) [“xyz”, “xzy”, “xxz”, “yzx”, “zyx”, “zxz”]

C) [“xyz”, “xzy”, “xxz”, “yzx”, “zyx”, “zxy”]

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Starting from the first index as the current index, recursively compute the permutations of the input string.

Compute the permutation by swapping the current index with every index in the remaining string.

Recurse the computation step by incrementing the current index by 1.

If we reach the end of the string, store the current string as a permutation.

Return the list of all permutations.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in the following coding playground:

Java

usercode > main.java

```java
import java.util.*;
public class Main{
    public static ArrayList<String> permuteWord(String word) {

        // Your code will replace this placeholder return statement

        return new ArrayList<>();
    }
}
```

Powered by AI | ▷ | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

"abcd"

Permutations

← Back

Solution: Subsets

Next →

Solution: Permutations

☑ Mark as Completed