Problem Challenge 1

Permutation in a String (hard)

Given a string and a pattern, find out if the **string contains any permutation of the pattern**.

Permutation is defined as the re-arranging of the characters of the string. For example, “abc” has the following six permutations:

1. abc
2. acb
3. bac
4. bca
5. cab
6. cba

If a string has ‘n’ distinct characters it will have $n!$ permutations.

Example 1:

```
Input: String="oidbcaf", Pattern="abc"
Output: true
Explanation: The string contains "bca" which is a permutation of the given pattern.
```

Example 2:

```
Input: String="odicf", Pattern="dc"
Output: false
Explanation: No permutation of the pattern is present in the given string as a substring.
```

Example 3:

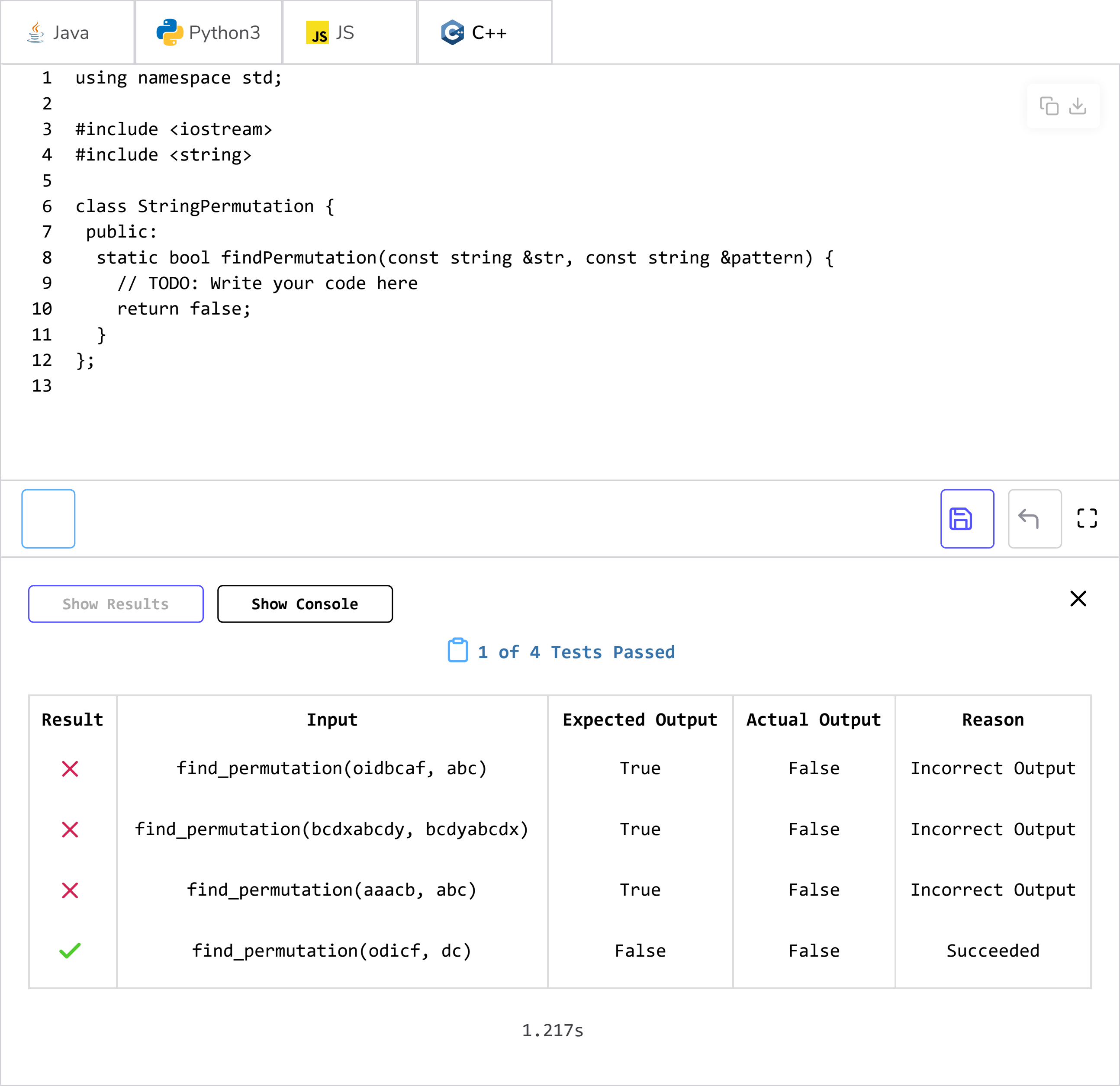
```
Input: String="bcdxabcdy", Pattern="bcdyabcdx"
Output: true
Explanation: Both the string and the pattern are a permutation of each other.
```

Example 4:

```
Input: String="aaacb", Pattern="abc"
Output: true
Explanation: The string contains "acb" which is a permutation of the given pattern.
```

Try it yourself

Try solving this question here:



```cpp
using namespace std;

#include <iostream>
#include <string>

class StringPermutation {
 public:
  static bool findPermutation(const string &str, const string &pattern) {
    // TODO: Write your code here
    return false;
  }
};

```

1 of 4 Tests Passed

Result	Input	Expected Output	Actual Output	Reason
✗	find_permutation(oidbcaf, abc)	True	False	Incorrect Output
✗	find_permutation(bcdxabcdy, bcdyabcdx)	True	False	Incorrect Output
✗	find_permutation(aaacb, abc)	True	False	Incorrect Output
✓	find_permutation(odicf, dc)	False	False	Succeeded

1.217s