

String Challenge

Have the function `StringChallenge(str)` take the `str` parameter being passed and determine if it is possible to create a palindromic string of minimum length 3 characters by removing 1 or 2 characters. For example: if `str` is "abjchba" then you can remove the characters `jc` to produce "abhba" which is a palindrome. For this example your program should return the two characters that were removed with no delimiter and in the order they appear in the string, so `jc`. If 1 or 2 characters cannot be removed to produce a palindrome, then return the string `not possible`.

If the input string is already a palindrome, your program should return the string `palindrome`. Your program should always remove the characters that appear earlier in the string. In the example above, you can also remove `ch` to form a palindrome but `jc` appears first, so the correct answer is `jc`.

The input will only contain lowercase alphabetic characters. Your program should always attempt to create the longest palindromic substring by removing 1 or 2 characters (see second sample test case as an example). The 2 characters you remove do not have to be adjacent in the string.

Examples

Input: "mmop"

Output: not possible

Input: "kjjjhjjj"

Output: k

insolent

Time left: 0 hours, 26 minutes

Array Challenge

Have the function `ArrayChallenge(arr)` read the array of integers stored in `arr` which will be in the following format: `[K, r1, r2, r3, ...]` where `K` represents the number of desks in a classroom, and the rest of the integers in the array will be in sorted order and will represent the desks that are already occupied. All of the desks will be arranged in 2 columns, where desk #1 is at the top left, desk #2 is at the top right, desk #3 is below #1, desk #4 is below #2, etc. Your program should return the number of ways 2 students can be seated next to each other. This means 1 student is on the left and 1 student on the right, or 1 student is directly above or below the other student.

For example, if `arr` is `[12, 2, 6, 7, 11]` then this classrooms looks like the following picture:



Based on above arrangement...

Java

```

1  import
2  import
3
4  class M
5
6  public
7  //
8  ret
9  }
10
11 public
12 //
13 Sca
14 Sys
15 }
16
17 }
    
```

.jar=52629:C:\Program Fil

Event Log

15:9