1. **Grid search: https://www.hackerrank.com/challenges/the-grid-search/problem (expected 15min)**
2. **Minimum cost (expected 20min)**

For a given staircase, the i-th step is assigned a non-negative cost indicated by a cost array.

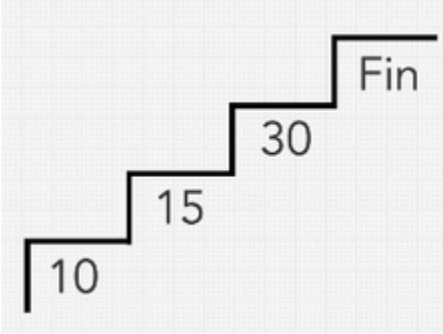
Once you pay the cost for a step, you can either climb one or two steps.

Find the minimum cost to reach the top of the staircase. Your first step can either be the first or second step.

Test cases:

1) Test case #1:

Input: [10, 15, 30]
Output: 15
The stair is:



2) Test case #2:

input: [10, 50, 2, 1, 8, 9]
output: 20

3) Test case #3:

input: [20, 15, 10, 30, 12, 4, 3]
output: 40

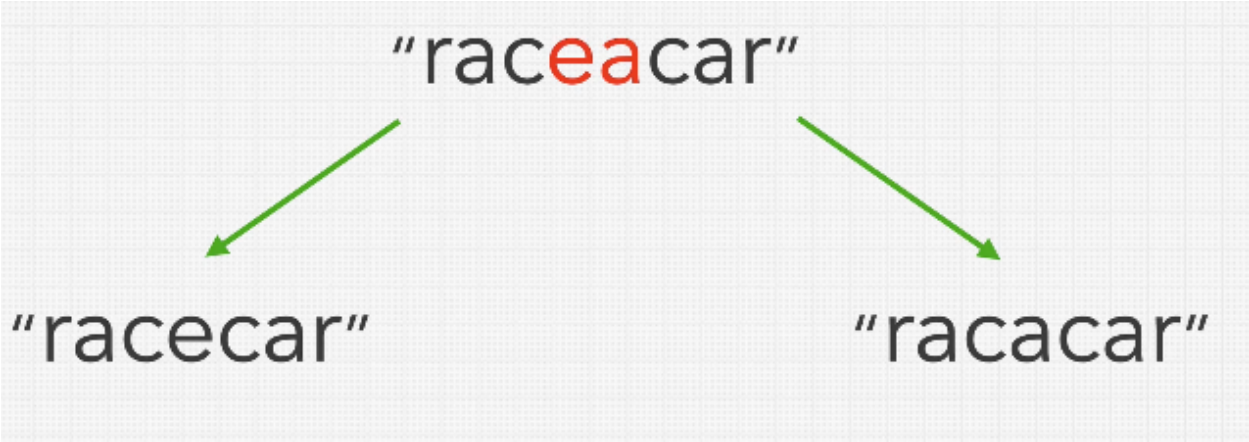
4) Hidden test case: [10, 20, 10, 5, 3, 1, 40, 80, 70]

3. **Almost a palindrome (expected 10min)**

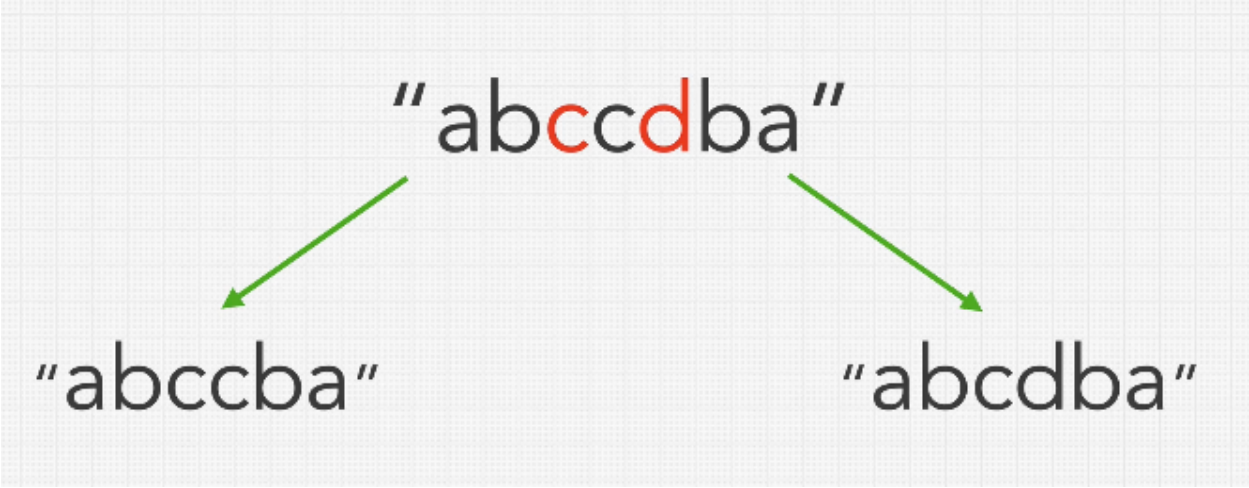
Given a string, determine if it is almost a palindrome.

A string is almost a palindrome if becomes a palindrome by removing 1 letter. Consider only alphanumeric characters and ignore case sensitivity

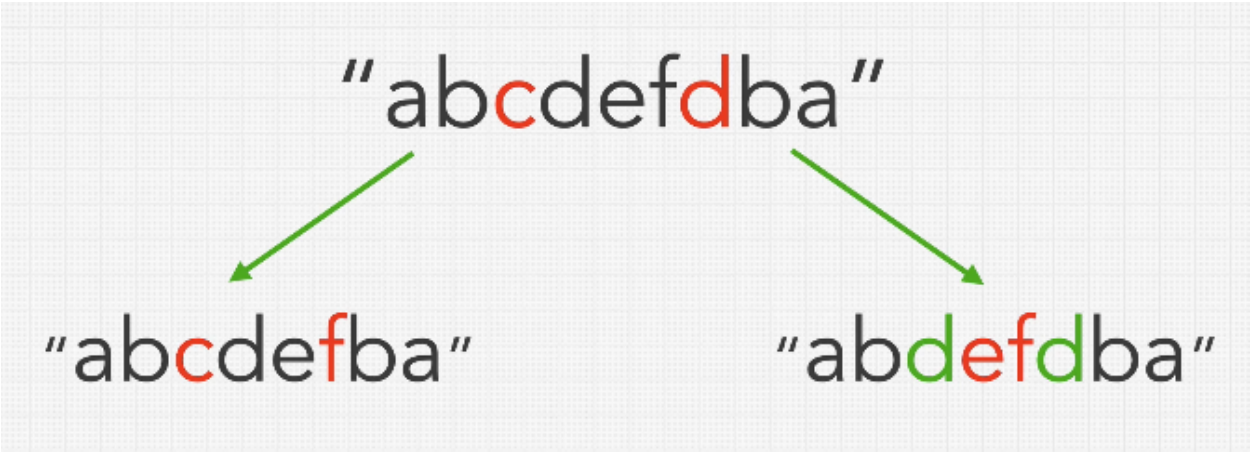
Examples:



Raceacar is almost a palindrome.- We can remove either the e or a



Abccdba is almost a palindrome only if we remove d



And abcdefdba is not almost a palindrome

Test cases:

1. Input: “raceacar” Output: True
2. Input: “Abccdba” Output: True
3. Input: “Abcdefdba” Output: False
4. Input: “” Output: True
5. Input: “A” Output: True
6. Input: “Ab” Output: True

4. Eerie Mob (expected 15min) - Medium en el Devsu

We all love emojis. ASCII emojis have existed for a while now, including this guy: **(-_-)**. You can represent a **mob** of these like this: `(-(-_(-_-(-_(-_(-_-)_-)_-)-_-)_-)-)`. **Looks scary!** That mob has 11 guys!

The rules to create a mob are as follows:

1. A mob can have **between 1 and 255 guys, 255 included**.
2. If there's an **even** number of guys, then there should be one more on the left than on the right: **eg 4:** `(-_(-_(-_-)_-)`
3. There are four different types of guys: a complete guy `(-_-)`, a side guy `_-)`, a partial guy `-_-)` and a final guy `-)`.
    a. The complete guy must be in the middle, and there must only be one per mob.
    b. A final guy must be on both sides, as long as there are **strictly more than 7 guys**.
    c. Let's say the **complete guy** is at the k^{th} position. Every $(k \pm 3n)^{th}$ guy must be a partial guy.
    d. All the rest are side guys.

An example, a mob that has 14 guys is this one.

`(-(-_-(-_(-_(-_-(-_(-_(-_-)_-)_-)-_-)_-)_-)-)`

Create a function that receives an `integer`, indicating how many guys are on your mob and returns a `string` with the resulting mob. If the parameter does not comply with the first rule, or if it's null or empty, then the function should return `(O_o)`