1- Given a list of numbers, create a function that returns a list where all similar adjacent elements have been reduced to a single element, so [1,2,3.3] returns [1,2,3]

Note:

You may create a new list or modify the passed in list.

2- Consider dividing a string into two halves

Case1:

The length is even, the front and back halves are the same length.

Case2:

The length is odd, we'll say that the extra char goes in the front half.

E.g. 'abced', the front half is 'abc', the back half'de.

Given 2 strings, a and b, return a string of the form:

(a-front + b-front) + (a-back +b-back)

3- Write a Python function that takes a sequence of numbers and determines

whether all the numbers are different from each other.

E.X. [1,5,7,9] -> True

[2,4,5,5,7,9] -> False

4- Given unordered list, sort it using algorithm bubble sort

(read about bubble sort and try to implement it)

5- Gusses game

- **Your game generates a random number and gives only 10 tries for the user to guess that number.**
- **Get a user input and compare it with the random number**
- **Display a hit message to the user in case the use number is smaller or bigger of the random number**
- **If the user type number is out of range(100), display a message that is not allowed and don't count this as try.**
- **If user type a number that has been entered before, display a hint message and don't count this as try**
- **In case the user entered a correct number within the 10 tries, display a congratulations message and let your application guess another random number with the remain number of tries**
- **If the user finishes all his tries, display a message to ask him if he wants to play again or not.**

6- Make account on Hacker-rank for problem solving
(bonus)
And try to solve this problem and send me your submission

https://www.hackerrank.com/challenges/diagonal-difference/problem