Hero’s quest board

Here is where we will share exercises we find so we can continue challenging monsters to become better heroes. Please, follow the structure below so that everyone can understand the exercises better.

Challenger: L
Estimation: [easy, medium, hard, very hard]
Challenge:
Text of the exercise to solve

Challenger:
Estimation: easy
Challenge: Given two strings, write a method to decide if one is a permutation of the other.

Challenger: Serife
Estimation: hard
Challenge: Given a string, write a function to check if it is a permutation of a palindrome. A palindrome is a word or phrase that is the same forwards and backwards. A permutation is a rearrangement of letters. The palindrome does not need to be limited to just dictionary words.
Example:
Input: Tact Coa
Output: True (permutations: "taco cat". "atco cta". etc.)

Challenger: Serife
Estimation: medium
Challenge: There are three types of edits that can be performed on strings: insert a character, remove a character, or replace a character. Given two strings, write a function to check if they are one edit (or zero edits) away.
Example
pale, ple-> true
pale, pales -> true
pale, bale -> true
pale, bake -> false

Challenger: Serife
Estimation: hard
Challenge:
Once upon a time, on a way through the old wild west, a man was given directions to go from one point to another. The directions were "NORTH", "SOUTH", "WEST", "EAST". Clearly "NORTH" and "SOUTH" are opposite, "WEST" and "EAST" too. Going to one direction and coming back the opposite direction is a needless effort. Since this is the wild west, with dreadful weather and not much water, it's important to save yourself some energy, otherwise you might die of thirst!

How I crossed the desert the smart way.

The directions given to the man are, for example, the following:

`["NORTH", "SOUTH", "SOUTH", "EAST", "WEST", "NORTH", "WEST"].`

You can immediately see that going "NORTH" and then "SOUTH" is not reasonable, better stay to the same place! So the task is to give to the man a simplified version of the plan. A better plan in this case is simply: [“WEST”]

Task : Write a function `dirReduc` which will take an array of strings and returns an array of strings with the needless directions removed (W<->E or S<->N side by side). Order of the unremoved directions should be protected. (Only the consecutive directions are allowed to be cancelled/removed.)

Example :
Input `["NORTH", "SOUTH", "EAST", "WEST"]`
`Output []`

`Input` ["NORTH", "EAST", "WEST", "SOUTH", "SOUTH", "WEST"]
Output ["SOUTH", "WEST"]

Challenger: Serife
Estimation: hard
Challenge:

Next smaller number with the same digits…..

Write a function that takes a positive integer and returns the next smaller positive integer containing the same digits.

Example :

```
nextSmaller(21) == 12
nextSmaller(531) == 513
nextSmaller(2071) == 2017
nextSmaller(513) == 351
nextSmaller(9670) == 9076
```

Return -1, when there is no smaller number that contains the same digits. Also return -1 when the next smaller number with the same digits would require the leading digit to be zero.

```
nextSmaller(9) == -1
nextSmaller(111) == -1
nextSmaller(135) == -1
nextSmaller(1027) == -1 // 0721 is out since we don't write numbers with leading 0.
```

Challenger: Serife
Estimation: easy
Challenge:
You have a 'Student' class as following:

```
public class Student {
  private final String firstName;
  private final String lastName;
  public final String studentNumber;
  public String getFullName() {
    return firstName + " " + lastName;
  }
  public String getCommaName() {
    return lastName + ", " + firstName;
  }
}
```

`Your task is to` write a Function (with the appropriate types) that returns true if a given student is "John Smith" with a student number of "js123" (otherwise return false).

Challenger: Serife
Estimation: medium
Challenge:

You have a ‘Triangle’ class as following:

```
public class Triangle {
  public final int height;
  public final int base;
  private double area;
  public void setArea(double a) {
    area = a;
  }
  public double getArea() {
    return area;
  }
}
```

Write a function that sets the area of the given Triangle and returns the area as a `double`.

Do not use the function type Function<T,R>; there is a function type for converting an arbitrary class into a `double`, use that.

The formula for triangle area is: 1/2 * base * height

Assume valid input (base and height are both greater than 0).

A full listing of the default function types can be found at

http://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html

Challenger: Serife
Estimation: medium
Challenge: Total amount of increasing or decreasing numbers up to a power of 10

Let's define `increasing` numbers as the numbers whose digits, read from left to right, are never less than the previous ones: 234559 is an example of increasing number.

Conversely, `decreasing` numbers have all the digits read from left to right so that no digits is bigger than the previous one: 97732 is an example of decreasing number.

You do not need to be the next Gauss to figure that all numbers with 1 or 2 digits are either increasing or decreasing: 00, 01, 02, ..., 98, 99 are all belonging to one of this categories (if not both, like 22 or 55): 101 is indeed the first number which does NOT fall into either of the categories. Same goes for all the numbers up to 109, while 110 is again a decreasing number.

Now your task is rather easy to declare (a bit less to perform): **you have to build a function to return the total occurrences of all the increasing or decreasing numbers *below* 10 raised to the xth power (x will always be >= 0).**

This means that your function will have to behave like this:

```
totalIncDec(0)==1
totalIncDec(1)==10
totalIncDec(2)==100
totalIncDec(3)==475
totalIncDec(4)==1675
totalIncDec(5)==4954
totalIncDec(6)==12952
```

Tips: efficiency and trying to figure out how it works are essential: with a brute force approach, some tests with larger numbers may take more than the total computing power currently on Earth to be finished in the short allotted time.

To make it even clearer, the increasing or decreasing numbers between in the range 101-200 are: [110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 122, 123, 124, 125, 126, 127, 128, 129, 133, 134, 135, 136, 137, 138, 139, 144, 145, 146, 147, 148, 149, 155, 156, 157, 158, 159, 166, 167, 168, 169, 177, 178, 179, 188, 189, 199, 200], that is 47 of them.

Challenger: Niklas
Estimation: medium

Challenge:

Given a time in 12-hour AM/PM format, convert it to military (24-hour) time.

Note:
Midnight is 12:00:00AM on a 12-hour clock, and 00:00:00 on a 24-hour clock.
Noon is 12:00:00PM on a 12-hour clock, and 12:00:00 on a 24-hour clock.

Function Description

The function should return a new string representing the input time in 24 hour format.

timeConversion has the following parameter(s):

- s: a string representing time in 12 hour format

Input Format

A single string containing a time in 12-hour clock format (i.e.: hh:mm:ssAM or hh:mm:ssPM), where and .

Prerequisites

- All input times are valid

Sample Input

07:05:45PM

Sample Output

19:05:45

Challenger: Niklas
Estimation: medium
Challenge:

Given the time in numerals we may convert it into words, as shown below:
At *minutes = 0*, use *o'clock*. For *minutes < 30*, use *past*, and for *minutes > 30* use *to*.

Description
The program should return a time string as described.
The method has the following parameter(s):

h: an integer representing hour of the day
m: an integer representing minutes after the hour

Sample Input 0: 5 , 47
Sample Output 0: thirteen minutes to six

Sample Input 1: 3 , 00
Sample Output 1: three o' clock

Road Service

Austrian rail-System decided to improved There are cities in IOI kingdom, numbered from

through

. There are also

bidirectional roads, numbered from

through

. The road

connects the city

and the city

. There exists a path between any pair of cities.

The distance between two cities are defined as the smallest number of roads which connect the two

cities. The total distance of IOI kingdom is defined as the sum of the distances between all pairs of

different cities.

The king of IOI kingdom plans to construct

additional roads in order to reduce the total distance

and improve convenience.

You, as an assistant of the king, help the king by finding a good plan.