

1. There is a car with *capacity* empty seats. The vehicle only drives east (i.e., it cannot turn around and drive west).

You are given the integer *capacity* and an array *trips* where *trips[i] = [numPassengers(i), from(i), to(i)]* indicates that the *i*th trip has *numPassengers(i)* passengers, and the locations to pick them up and drop them off are *from(i)* and *to(i)* respectively. The locations are given as the number of kilometers due east from the car's initial location.

Return true if it is possible to pick up and drop off all passengers for all the given trips, or false otherwise.

Example 1:

Input: *trips* = [[2,1,5],[3,3,7]], *capacity* = 4

Output: false

Example 2:

Input: *trips* = [[2,1,5],[3,3,7]], *capacity* = 5

Output: true

2. Given a string *s*, rearrange the characters of *s* so that any two adjacent characters are not the same.

Return any possible rearrangement of *s* or return "" if not possible.

Example 1:

Input: *s* = "aab"

Output: "aba"

Example 2:

Input: *s* = "aaab"

Output: ""

3. Determine if a 9 x 9 Sudoku board is valid. Only the filled cells need to be validated according to the following rules:

Each row must contain the digits 1-9 without repetition.

Each column must contain the digits 1-9 without repetition.

Each of the nine 3 x 3 sub-boxes of the grid must contain the digits 1-9 without repetition.

Note:

A Sudoku board (partially filled) could be valid but is not necessarily solvable.

Only the filled cells need to be validated according to the mentioned rules.

Example 1:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Input: board =

```
[["5","3",".",".","7",".",".",".","."],
["6",".",".","1","9","5",".",".","."],
[".","9","8",".",".",".","6","."],
["8",".",".","6",".",".","3"],
["4",".","8","3",".","1"],
["7",".","2",".","6"],
[".","6",".","2","8","."],
[".","4","1","9",".","5"],
[".","8",".","7","9"]]
```

Output: true

Example 2:

Input: board =

```
[["8","3",".",".","7",".",".",".","."],
["6",".",".","1","9","5",".",".","."],
[".","9","8",".",".",".","6","."],
["8",".","6",".","3"],
["4",".","8","3",".","1"],
["7",".","2",".","6"],
[".","6",".","2","8","."],
[".","4","1","9",".","5"],
[".","8",".","7","9"]]
```

Output: false

Explanation: Same as Example 1, except with the 5 in the top left corner being modified to 8. Since there are two 8's in the top left 3x3 sub-box, it is invalid.

4. You are given a string s and an integer k . You can choose any character of the string and change it to any other uppercase English character. You can perform this operation at most k times.

Return the length of the longest substring containing the same letter you can get after performing the above operations.

Example 1:

Input: $s = \text{"ABAB"}, k = 2$

Output: 4

Explanation: Replace the two 'A's with two 'B's or vice versa.

Example 2:

Input: $s = \text{"AABABBA"}, k = 1$

Output: 4

Explanation: Replace the one 'A' in the middle with 'B' and form "AABBBBA".

The substring "BBBB" has the longest repeating letters, which is 4.

There may exist other ways to achieve this answer too.