

Composed of three of us in the early October

===== Tester #1 =====

(1)

☆ Usernames System

In this challenge you will create the username portion of a registration system. All usernames must be unique. If a new user requests a name that is already used, an integer should be added to the end of the username to make it unique. The numbering begins with 1 and is incremented by 1 for each new instance per username.

As an example, if username requests were for [bob, ,alice, bob, alice, bob, alice], your system should assign usernames [bob, alice, bob1, alice1, bob2, alice2].

Given a list of username requests in the order given, process all requests and return an array of the usernames as assigned by your function.

Function Description

Complete the function `usernamesSystem` in the editor below. The function must return an array of usernames in the order assigned.

`usernamesSystem` has the following parameter(s):

`u[u0...un-1]`: an array of username strings in the order requested

Constraints

- $1 \leq n \leq 10^4$
- $1 \leq |u_i| \leq 20$
- u_i contains only lowercase English letters in the range `ascii[a-z]`.

▼ Sample Case 0

Sample Input

```
4
alex
xylos
alex
alan
```

Sample Output 0

```
alex
xylos
alex1
alan
```

解釋：偵測使用者帳號是否重複的，如果有重複的在後面加上integer number，比如第一次輸入alex沒問題，第二次重複輸入alex就要改成alex1輸出，再有重複就要改成alex2輸出。注意題目的限制比較寬鬆，所有的輸入字串只會有英文[a-z]。

(2)

☆ Minimum Unique Array Sum

Given an array, you must increment any duplicate elements until all its elements are unique. In addition, the sum of its elements must be the minimum possible within the rules. For example, if `arr = [3, 2, 1, 2, 7]`, then `arrunique = [3, 2, 1, 4, 7]` and its elements sum to a minimal value of $3 + 2 + 1 + 4 + 7 = 17$.

Function Description

Complete the `getMinimumUniqueSum` function in the editor below to create an array of unique elements with a minimal sum. Return the integer sum of the resulting array.

`getMinimumUniqueSum` has the following parameter(s):

`arr`: an array of integers to process

Constraints

- $1 \leq n \leq 2000$
- $1 \leq arr[i] \leq 3000$ where $0 \leq i < n$

▼ Sample Case 0

Sample Input 0

```
3
1
2
2
```

Sample Output 0

```
6
```

Explanation 0

`arr = [1, 2, 2]`

The duplicate array elements 2 must be addressed. The minimum unique array will be achieved by incrementing one of the twos by 1, creating the array [1, 2, 3]. The sum of elements in the new array is $1 + 2 + 3 = 6$.

解釋：給定一個整數陣列，如果有重複的element，持續+1直到沒有重複，最後輸出整個陣列的總和 ($O(n^2)$ 很容易，但是有 $O(n\log n)$ 解)

Making elements distinct in a sorted array by minimum increments

<https://www.geeksforgeeks.org/making-elements-distinct-sorted-array-minimum-increments/>

===== Tester #2 =====

(1)

☆ Shifting Strings

We define the following operations on a string:

- *Left Shift*: A single circular rotation of the string in which the first character becomes the last character and all other characters are shifted one index to the left. For example, *abcde* becomes *bcdea* after one left shift and *cdeab* after two left shifts.
- *Right Shift*: A single circular rotation of the string in which the last character becomes the first character and all other characters are shifted one index to the right. For example, *abcde* becomes *eabcd* after one right shift and *deabc* after two right shifts.

Function Description

Complete the function *getShiftedString* in the editor below. The function must return the string *s* after performing the stated shifts.

getShiftedString has the following parameter(s):

s: the string to shift
leftShifts: integer
rightShifts: integer

Constraints

- $1 \leq |s| \leq 10^5$
- $0 \leq \text{leftShifts}, \text{rightShifts} \leq 10^9$
- String *s* consists of lowercase English alphabetic letters only, `ascii[a-z]`.

解釋：字串向左rotate和向右rotate，用兩個相同字串接再一起可以簡單解出

(2)

☆ Distinct Pairs

In this challenge, you will be given an array of integers and a target value. Determine the number of *distinct* pairs of elements in the array that sum to the target value. Two pairs (a, b) and (c, d) are considered to be distinct if and only if the values in sorted order do not match, i.e., (1, 9) and (9, 1) are indistinct but (1, 9) and (9, 2) are distinct.

For instance, given the array [1, 2, 3, 6, 7, 8, 9, 1], and a target value of 10, the seven pairs (1,9), (2,8), (3,7), (8, 2), (9, 1), (9, 1), and (1, 9) all sum to 10 and only three distinct pairs: (1, 9), (2, 8), and (3, 7).

Function Description

Complete the function *numberOfPairs* in the editor below. The function must return an integer, the total number of *distinct* pairs of elements in the array that sum to the target value.

numberOfPairs has the following parameter(s):

a[a[0],...,a[n-1]]: an array of integers to select pairs from
k: target integer value to sum to

Constraints

- $1 \leq n \leq 5 \times 10^5$
- $0 \leq a[i] \leq 10^9$
- $0 \leq k \leq 5 \times 10^9$

解釋：就是two sum problem，不過最後要輸出符合的pair數量，同樣的pair只能算一次 (如果有{1,9} {9,1}，在最後的count數只能+1)，注意題目給的陣列會出現重複的element

===== Tester #3 =====

(1)

☆ Can You Sort?

An array of integers *arr*, of size *n* is defined as *a[0], a[1], ..., a[n-1]*. You will be given an array of integers to sort. Sorting must first be by frequency of occurrence, then by value. For instance, given an array [4, 5, 6, 5, 4, 3], there is one each of 6's and 3's, and there are two 4's, two 5's. The sorted list is [3, 6, 4, 4, 5, 5].

Function Description

Complete the function *customSort* in the editor below. The function must print the array each element on a separate line, sorted ascending first by frequency of occurrence, then by value within frequency.

customSort has the following parameter(s):

arr[arr[0],...,arr[n-1]]: an array of integers to sort

Constraints

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq arr[i] \leq 10^6$

解釋：先根據element出現的次數sort，如果出現次數一樣再根據大小sort

(2)

☆ The Social Network

A social network has n active users, numbered from 0 to $n - 1$, who selectively friend other users to create groups of friends within the network. We define the following:

- Two users, x and y , are *direct* friends if they friend each other on the network.
- Two users, x and z , are *indirect* friends if there exists some direct friend, y , common to both users x and z .
- Two users, x and y , belong to the same *group* if they are friends (either directly or indirectly) with each other. In other words, if user x is part of a group, then all of user x 's friends and friends of friends belong to the same group.
- We describe the number of people in each group as an array of n integers, *counts*, where each *counts* _{i} ($0 \leq i < n$) denotes the total number of users in the group that user i belongs to. For example, if *counts* = [3, 3, 3, 3, 3, 1, 3], then there are three groups; users 0, 1, 2, 3, 4, and 6 are in one of two 3-person groups, and user 5 is in a 1-person group.

ID	0	1	2	3	4	5	6
Group Size	3	3	3	3	3	1	3

- A group is *valid* if all the users in the group have minimal ID numbers. In other words, a group of size k must contain the k smallest ID numbers belonging to a group of that size with respect to the smallest user ID in the group. For example, if *counts* = [3, 3, 3, 3, 3, 1, 3], then the grouping [0, 1, 2], [3, 4, 6], and [5] is valid; however, the grouping [0, 1, 4], [2, 3, 6], and [5] is *not* valid because the group [0, 1, 4] does not contain the three smallest user IDs for the set of user IDs belonging to 3-person groups (i.e., {0, 1, 2, 3, 4, 6}).

► Example

- We print the information for each valid group on a new line in the format **user_{smallest ID} ... user_{largest ID}**, where the users within each group are ordered by ascending ID and the groups themselves are ordered by ascending *user_{smallest ID}*. For example, we print the valid grouping [0, 1, 2], [3, 4, 6], and [5] for *counts* = [3, 3, 3, 3, 3, 1, 3] as:

輸出：

0 1 2 (同為size = 3的group)

3 4 6 (同為size = 3的group)

5 (同為size = 1的group)

▼ Sample Case 0

Sample Input

```
4
2
2
2
2
```

Sample Output

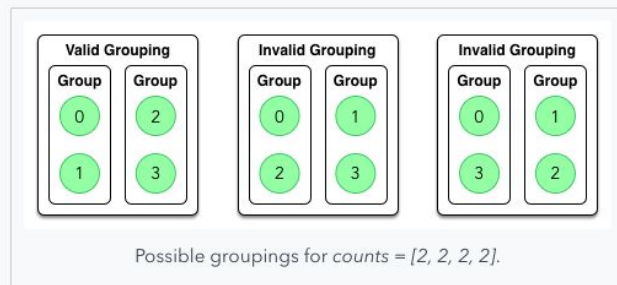
```
0 1
2 3
```

Explanation

We express *counts* = [2, 2, 2, 2] as the following table of group sizes:

ID	0	1	2	3
Group Size	2	2	2	2

The *valid* grouping here is the groups [0, 1] and [2, 3]:



解釋：一開始會給你每個element所屬的group size number，逐行輸出每個group的成員(注意同一行內的element要升冪排列，行與行之間的第一個element也要升冪排列)