UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA FACULTAD DE PRODUCCION Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Curso: Laboratorio de Análisis y Diseño de Algoritmos

Aula 11

Presentado por:

Tacca Apaza, Nohelia Estefhania

Docente:

Alex Josue Florez Farfan

Grupo-"B"

Arequipa - Perú

Diciembre 2021

Ejercicio 01 - Group Anagrams

Enlace: https://leetcode.com/problems/group-anagrams/

Given an array of strings strs, group the anagrams together. You can return the answer in any order.

An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Example 1:

```
Input: strs = ["eat","tea","tan","ate","nat","bat"]
Output: [["bat"],["nat","tan"],["ate","eat","tea"]]
```

Example 2:

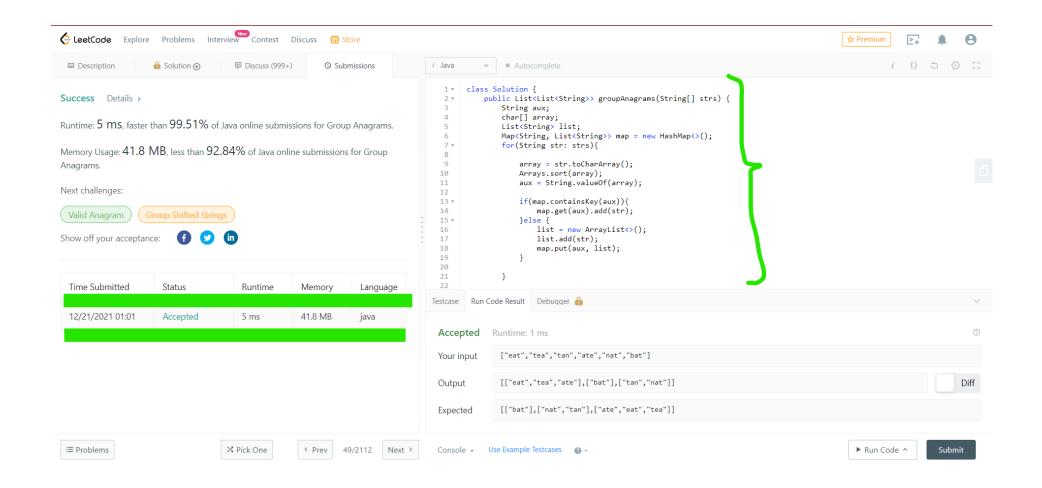
```
Input: strs = [""]
Output: [[""]]
```

Example 3:

```
Input: strs = ["a"]
Output: [["a"]]
```

Constraints:

- 1 <= strs.length <= 10⁴
- 0 <= strs[i].length <= 100
- strs[i] consists of lowercase English letters.



Ejercicio 02 - Dyslectionary

Enlace: https://open.kattis.com/problems/dyslectionary

Have you ever been uncertain how to spell a word? If you ask someone, they will sometimes just tell you to look it up, and that's fine if you know how the word starts. However, it's not much help if you only feel certain about how the word ends. To help give a equal treatment to words that are easy to spell at the end, you have developed the DyslectionaryTM. The Dyslectionary is just like an ordinary dictionary except that it organizes words based on how they end rather than how they start. The first section covers words ending with 'a'. Within this section words are ordered by their second-to-last letter, and so on. If two words have the same suffix, the shorter word is sorted earlier. To help organize your Dyslectionary, you are developing a simple computer program to harvest words from ordinary dictionaries.

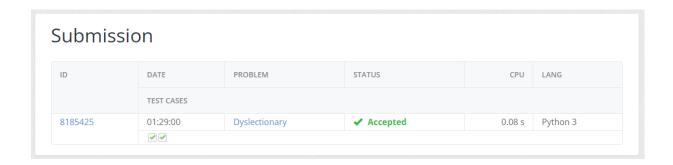
Input

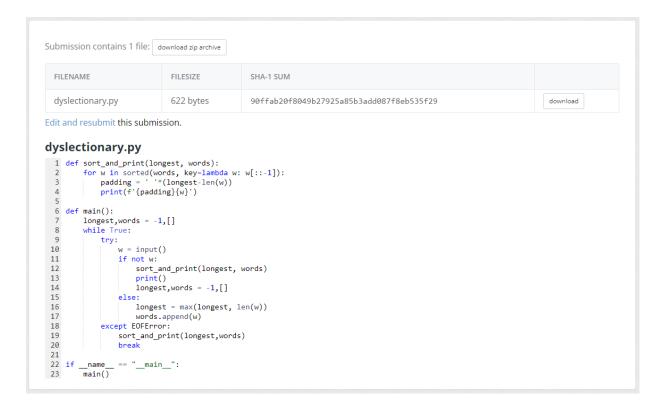
Input contains up to 100 word groups. Each word group contains up to 100 words, one word per line, listed in regular dictionary order. All words contain only lowercase English letters a – z. A blank line separates each pair of consecutive groups.

Output

For each group, print out its words in Dyslectionary order, one word per line. Use a blank line to separate consecutive groups. To make words easy to lookup, right-justify them so that they all end at the same column and there is just enough space to fit all the words.

Sample Input 1 Sample Output 1 apple banana banana apple grape grape kiwi kiwi pear pear airplane bicycle bicycle airplane boat car boat





Ejercicio 02 - Finding Borders

Enlace: https://cses.fi/problemset/task/1732/

A border of a string is a prefix that is also a suffix of the string but not the whole string. For example, the borders of abcabacab are ab and abcab.

Your task is to find all border lengths of a given string.

Input

The only input line has a string of length n consisting of characters a–z.

Output

Print all border lengths of the string in increasing order. Constraints

 $1 \le n \le 106$

Example

Input:

abcababcab

Output:

25

CSES Problem Set

Finding Borders

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

Submission details

Task:	Finding Borders		
Sender:	Nohelia		
Submission time:	2021-12-21 04:16:34		
Language:	C++11		
Status:	READY		
Result:	ACCEPTED		

Test results A

test	verdict	time	
#1	ACCEPTED	0.01 s	<u>>></u>
#2	ACCEPTED	0.01 s	<u>>></u>
#3	ACCEPTED	0.01 s	<u>>></u>
#4	ACCEPTED	0.01 s	<u>>></u>
#5	ACCEPTED	0.01 s	<u>>></u>
#6	ACCEPTED	0.12 s	<u>>></u>
#7	ACCEPTED	0.08 s	<u>>></u>
#8	ACCEPTED	0.03 s	<u>>></u>
#9	ACCEPTED	0.02 s	<u>>></u>
#10	ACCEPTED	0.01 s	<u>>></u>
#11	ACCEPTED	0.03 s	<u>>></u>

String Algorithms Word Combinations -String Matching _ Finding Borders **~** Finding Periods -Minimal Rotation -Longest Palindrome -Required Substring _ Palindrome Queries -Your submissions 2021-12-21 04:16:34

CSES Problem Set

Finding Borders

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

Number of submissions: 1

< 1 >

time	lang	code time	code size	result	
2021-12-21 04:16:34	C++	0.12 s	501 ch.	>	<u>>></u>

String Algorithms Word Combinations -String Matching -Finding Borders **~** Finding Periods -Minimal Rotation -Longest Palindrome -Required Substring -Palindrome Queries _ Your submissions 2021-12-21 04:16:34 **V**

Code A

```
1 #include <bits/stdc++.h>
3 using namespace std;
4 typedef long long 11;
5 const ll MOD = 1e9+7;
6 const ll p1 = 31;
7 const 11 p2 = 37;
8 const int maxN = 1e6+5;
10 int N;
11 | 11 pow1[maxN], pow2[maxN], ph1, ph2, sh1, sh2;
12 char S[maxN];
13
14 int main(){
15 scanf("%s", S);
16
        N = (int) strlen(S);
17
18
        pow1[0] = pow2[0] = 1;
        for(int i = 1; i < N; i++){}
19
            pow1[i] = (pow1[i-1] * p1) % MOD;
pow2[i] = (pow2[i-1] * p2) % MOD;
20
21
22
23
24
        for(int i = 0; i < N-1; i++){
            int l = (S[i] - 'a' + 1);
int r = (S[N-i-1] - 'a' + 1);
25
26
27
            ph1 = (ph1 + 1 * pow1[i]) % MOD;
28
            ph2 = (ph2 + 1 * pow2[i]) % MOD;
29
30
            sh1 = (sh1 * p1 + r) \% MOD;
            sh2 = (sh2 * p2 + r) % MOD;
31
32
33
            if(ph1 == sh1 && ph2 == sh2)
                 printf("%d ", i+1);
34
35
        }
36
```