

**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

## **Práctica 12**

**Estudiante:** Tapara Quispe, Fabiola Grissel

**Docente:** Alex Josue Florez Farfan

**Sección:** “B”


Arequipa - Perú

Diciembre 2021

## Ejercicio 2

open.kattis.com/submissions/8185533

UNSA 20 M V D T GD apa SC S f j N GA WhatsApp b b2 b3 I CSS a BST C



**Kattis**  
PROBLEMS CONTESTS RANKLISTS JOBS (5) HELP

Search Kattis

Submit

FABIOLA GRISEL TAP...  
Score: 12.2, Rank: 44083

### Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
TEST CASES					
8185533	20:12:27	Dyslectionary	✓ Accepted	0.00 s	C++
✓✓					

Submission contains 1 file: [download zip archive](#)




FILENAME	FILESIZE	SHA-1 SUM	
Ejercicio2.cpp	1418 bytes	04dfbe1db281e24fdee9687b5747a89f23c14fd9	<a href="#">download</a>

[Help](#) [edit and resubmit this submission.](#)

## Ejercicio2.cpp

```
1 #include <bits/stdc++.h>
2 /*
3  * Ejercicio 2 Dyslectionary
4  * Autor: Fabiola Tapara Quispe
5  * Descripcion: La distilencia es como un diccionario ordinario,
6  * excepto que organiza palabras basadas en cómo terminan en lugar
7  * de cómo comienzan. Si dos palabras tienen el mismo sufijo, la
8  * palabra más corta se ordena anteriormente
9  * Fecha: 20/12/21
10 */
11 #include <algorithm>
12 #include <iostream>
13 #include <vector>
14 #include <string>
15
16 using namespace std;
17
18 void process(vector<string>& v) {
19     // Encuentra la longitud máxima para cada cadena
20     int maxlen = 0;
21     for(auto i : v) {
22         if(i.length() > maxlen) {
23             maxlen = i.length();
24         }
25     }
26
27     // Invierta cada cadena, ordena, voltea hacia atrás.
28     for(auto& i : v) {
29         reverse(i.begin(), i.end());
30     }
31     sort(v.begin(), v.end());
32     for(auto& i : v) {
33         reverse(i.begin(), i.end());
34     }
35
36     // Pad todas las cuerdas con espacios.
37     for(auto& i : v) {
38         string temp;
39         temp.resize(maxlen - i.length(), ' ');
40         temp += i;
```

## Ejercicio 4

 FabiolaTapara —  

**CSES Problem Set**  
**Finding Borders**

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

**Submission details**  
Task: [Finding Borders](#)  
Sender: FabiolaTapara  
Submission time: 2021-12-21 03:13:00  
Language: C++11  
Status: READY  
Result: **ACCEPTED**

**Test results** ▲

test	verdict	time	
#1	ACCEPTED	0.01 s	»»
#2	ACCEPTED	0.01 s	»»
#3	ACCEPTED	0.01 s	»»
#4	ACCEPTED	0.01 s	»»
#5	ACCEPTED	0.01 s	»»
#6	ACCEPTED	0.10 s	»»
#7	ACCEPTED	0.06 s	»»
#8	ACCEPTED	0.01 s	»»
#9	ACCEPTED	0.01 s	»»

**String Algorithms**  
[Word Combinations](#) —  
[String Matching](#) —  
[Finding Borders](#) ✓  
[Finding Periods](#) —  
[Minimal Rotation](#) —  
[Longest Palindrome](#) —  
[Required Substring](#) —  
[Palindrome Queries](#) —  
...  
**Your submissions**  
2021-12-21 03:13:00 ✓

#10	ACCEPTED	0.01 s	>>
#11	ACCEPTED	0.02 s	>>

### Code ▲

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 /*
4  * Ejercicio 4 Finding Borders
5  * Autor: Fabiola Tapara Quispe
6  * Descripcion: Un borde de una cadena es un prefijo que también
7  * es un sufixo de la cadena, pero no la cadena entera. Por ejemplo,
8  * los bordes de "Abcababcbab" son "AB" y "ABCAB". Su tarea es encontrar todos
9  * las longitudes de borde de una cadena dada.
10  * Fecha: 20/12/21
11  */
12
13 int main() {
14     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
15     string s; cin >> s;
16     int n = s.length();
17     vector<int> pi(n), A;
18     for (int i = 1, j = 0; i < n; ++i) {
19         while (j > 0 && s[i] != s[j]) j = pi[j - 1];
20         if (s[i] == s[j]) ++j;
21         pi[i] = j;
22     }
23     int j = pi[n - 1];
24     while (j > 0) {
25         A.emplace_back(j);
26         j = pi[j - 1];
27     }
28     reverse(A.begin(), A.end());
29     for (int len: A) {
30         cout << len << " ";
31     }
32     cout << "\n";
33     return 0;
34 }

```