

CodeForge - B01 - Độ Dài Chuỗi

Độ khó: ★ Easy

Đề bài

Viết chương trình đếm số ký tự trong chuỗi (sử dụng `length()`).

◇ Input

- Một dòng chứa chuỗi (có thể chứa khoảng trắng)

◇ Output

- In ra độ dài của chuỗi

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
11
```

Test case 2

Input:

```
Java
```

Output:

```
4
```

Test case 3

Input:**Output:**

0

Tags: `string`, `length`, `basic`

CodeForge - B02 - Kiểm Tra Chuỗi Rỗng

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có rỗng không (sử dụng `isEmpty()`).

◇ Input

- Một dòng chứa chuỗi (có thể rỗng hoặc chỉ chứa khoảng trắng)

◇ Output

- In ra **EMPTY** nếu chuỗi rỗng (độ dài = 0)
- In ra **NOT EMPTY** nếu chuỗi không rỗng

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

Hello

Output:

NOT EMPTY

Test case 2

Input:

EMPTY

Output:

Test case 3

Input:**Output:**

NOT EMPTY

Giải thích: Chuỗi chứa khoảng trắng nên không rỗng

Tags: `string`, `isEmpty`, `validation`

CodeForge - B03 - Kiểm Tra Chuỗi Blank

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có blank không (rỗng hoặc chỉ chứa khoảng trắng - sử dụng `isBlank()` - Java 11+).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra **BLANK** nếu chuỗi rỗng hoặc chỉ chứa khoảng trắng
- In ra **NOT BLANK** nếu chuỗi có ký tự thực sự

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

Hello

Output:

NOT BLANK

Test case 2

Input:

BLANK

Output:

Test case 3

Input:**Output:**

BLANK

Test case 4

Input:

Hi

Output:

NOT BLANK

Tags: `string`, `isBlank`, `validation`, `java11`

CodeForge - B04 - Lấy Ký Tự Tại Vị Trí K

Độ khó: ★ Easy

Đề bài

Lấy ký tự tại vị trí K trong chuỗi (sử dụng `charAt()`).

◇ Input

- Dòng 1: Một chuỗi
- Dòng 2: Số nguyên không âm K (vị trí, 0-indexed)

◇ Output

- In ra ký tự tại vị trí K
- Nếu K nằm ngoài phạm vi, in ra `INVALID`

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$
- $0 \leq K \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello
2
```

Output:

```
l
```

Test case 2

Input:

```
Java
0
```

Output:

```
J
```

Test case 3

Input:

```
Hi  
5
```

Output:

```
INVALID
```

Tags: `string`, `charAt`, `indexing`

CodeForge - B05 - Tìm Vị Trí Đầu Tiên

Độ khó: ★ Easy

Đề bài

Tìm vị trí xuất hiện đầu tiên của một ký tự trong chuỗi (sử dụng `indexOf()`).

◇ Input

- Dòng 1: Một chuỗi
- Dòng 2: Một ký tự cần tìm

◇ Output

- In ra vị trí đầu tiên (0-indexed)
- Nếu không tìm thấy, in ra `-1`

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
o
```

Output:

```
4
```

Test case 2

Input:

```
Java Programming
a
```

Output:

```
1
```

Test case 3

Input:

```
Hello  
x
```

Output:

```
-1
```

Tags: `string`, `indexOf`, `search`

CodeForge - B06 - Tìm Vị Trí Cuối Cùng

Độ khó: ★ Easy

Đề bài

Tìm vị trí xuất hiện cuối cùng của một ký tự trong chuỗi (sử dụng `lastIndexOf()`).

◇ Input

- Dòng 1: Một chuỗi
- Dòng 2: Một ký tự cần tìm

◇ Output

- In ra vị trí cuối cùng (0-indexed)
- Nếu không tìm thấy, in ra `-1`

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
o
```

Output:

```
7
```

Test case 2

Input:

```
Java Programming
a
```

Output:

```
12
```

Test case 3

Input:

```
Hello  
x
```

Output:

```
-1
```

Tags: `string`, `lastIndexOf`, `search`

CodeForge - B07 - Cắt Chuỗi Con

Độ khó: ★ ★ Medium

Đề bài

Cắt chuỗi con từ vị trí **start** đến **end** (sử dụng **substring()**).

◇ Input

- Dòng 1: Một chuỗi
- Dòng 2: Hai số nguyên **start** và **end** (0-indexed)

◇ Output

- In ra chuỗi con từ vị trí **start** đến **end-1**
- Nếu không hợp lệ, in ra **INVALID**

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$
- $0 \leq \text{start}, \text{end} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
0 5
```

Output:

```
Hello
```

Test case 2

Input:

```
Java Programming
5 16
```

Output:

```
Programming
```

Test case 3

Input:

```
Hello  
2 2
```

Output:**Giải thích:** Chuỗi rỗng**Tags:** `string`, `substring`, `slicing`

CodeForge - B08 - Chuyển Thành Chữ Thường

Độ khó: ★ Easy

Đề bài

Chuyển tất cả ký tự trong chuỗi thành chữ thường (sử dụng `toLowerCase()`).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra chuỗi sau khi chuyển thành chữ thường

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
hello world
```

Test case 2

Input:

```
JAVA PROGRAMMING
```

Output:

```
java programming
```

Test case 3

Input:

123AbC

Output:

123abc

Tags: `string`, `toLowerCase`, `case-conversion`

CodeForge - B09 - Chuyển Thành Chữ Hoa

Độ khó: ★ Easy

Đề bài

Chuyển tất cả ký tự trong chuỗi thành chữ hoa (sử dụng `toUpperCase()`).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra chuỗi sau khi chuyển thành chữ hoa

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
HELLO WORLD
```

Test case 2

Input:

```
java programming
```

Output:

```
JAVA PROGRAMMING
```

Test case 3

Input:

123AbC

Output:

123ABC

Tags: `string`, `toUpperCase`, `case-conversion`

CodeForge - B10 - Xóa Khoảng Trắng Đầu Cuối

Độ khó: ★ Easy

Đề bài

Xóa khoảng trắng ở đầu và cuối chuỗi (sử dụng `trim()`).

◇ Input

- Một dòng chứa chuỗi (có thể có khoảng trắng đầu cuối)

◇ Output

- In ra chuỗi sau khi trim

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
Hello World
```

Test case 2

Input:

```
Java
```

Output:

```
Java
```

Test case 3

Input:

Output:

Tags: `string`, `trim`, `whitespace`

CodeForge - B11 - So Sánh Chuỗi (equals)

Độ khó: ★ Easy

Đề bài

So sánh hai chuỗi (sử dụng `equals()`).

◇ Input

- Dòng 1: Chuỗi thứ nhất
- Dòng 2: Chuỗi thứ hai

◇ Output

- In ra `EQUAL` nếu hai chuỗi giống nhau
- In ra `NOT EQUAL` nếu khác nhau

◇ Constraints

- $0 \leq \text{độ dài mỗi chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello
Hello
```

Output:

```
EQUAL
```

Test case 2

Input:

```
Hello
hello
```

Output:

```
NOT EQUAL
```

Test case 3

Input:

```
Java  
Java
```

Output:

```
NOT EQUAL
```

Giải thích: Chuỗi thứ 2 có khoảng trắng cuối

Tags: `string`, `equals`, `comparison`

CodeForge - B12 - So Sánh Không Phân Biệt Hoa Thường

Độ khó: ★ Easy

Đề bài

So sánh hai chuỗi không phân biệt hoa thường (sử dụng `equalsIgnoreCase()`).

◇ Input

- Dòng 1: Chuỗi thứ nhất
- Dòng 2: Chuỗi thứ hai

◇ Output

- In ra `EQUAL` nếu hai chuỗi giống nhau (không phân biệt hoa thường)
- In ra `NOT EQUAL` nếu khác nhau

◇ Constraints

- $0 \leq \text{độ dài mỗi chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello  
hello
```

Output:

```
EQUAL
```

Test case 2

Input:

```
JAVA  
java
```

Output:

```
EQUAL
```

Test case 3

Input:

```
Hello  
Hi
```

Output:

```
NOT EQUAL
```

Tags: `string`, `equalsIgnoreCase`, `comparison`

CodeForge - B13 - So Sánh Từ Điển (compareTo)

Độ khó: ★ ★ Medium

Đề bài

So sánh hai chuỗi theo thứ tự từ điển (sử dụng `compareTo()`).

◇ Input

- Dòng 1: Chuỗi thứ nhất
- Dòng 2: Chuỗi thứ hai

◇ Output

- In ra **BEFORE** nếu chuỗi 1 đứng trước chuỗi 2 (< 0)
- In ra **AFTER** nếu chuỗi 1 đứng sau chuỗi 2 (> 0)
- In ra **EQUAL** nếu hai chuỗi bằng nhau ($= 0$)

◇ Constraints

- $1 \leq \text{độ dài mỗi chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
apple  
banana
```

Output:

```
BEFORE
```

Test case 2

Input:

```
zebra  
apple
```

Output:

```
AFTER
```

Test case 3

Input:

```
Hello  
Hello
```

Output:

```
EQUAL
```

Tags: `string`, `compareTo`, `lexicographic`

CodeForge - B14 - Kiểm Tra Chứa Chuỗi Con

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có chứa chuỗi con hay không (sử dụng `contains()`).

◇ Input

- Dòng 1: Chuỗi chính
- Dòng 2: Chuỗi con cần kiểm tra

◇ Output

- In ra **YES** nếu chuỗi chính chứa chuỗi con
- In ra **NO** nếu không chứa

◇ Constraints

- $1 \leq \text{độ dài chuỗi chính} \leq 10^5$
- $1 \leq \text{độ dài chuỗi con} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
World
```

Output:

```
YES
```

Test case 2

Input:

```
Java Programming
Python
```

Output:

NO

Test case 3

Input:

abcdef
cde

Output:

YES

Tags: `string`, `contains`, `search`

CodeForge - B15 - Kiểm Tra Bắt Đầu Bằng

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có bắt đầu bằng tiền tố cho trước không (sử dụng `startsWith()`).

◇ Input

- Dòng 1: Chuỗi chính
- Dòng 2: Tiền tố cần kiểm tra

◇ Output

- In ra **YES** nếu chuỗi bắt đầu bằng tiền tố
- In ra **NO** nếu không

◇ Constraints

- $1 \leq \text{độ dài chuỗi chính} \leq 10^5$
- $1 \leq \text{độ dài tiền tố} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
Hello
```

Output:

```
YES
```

Test case 2

Input:

```
Java Programming
Python
```

Output:

NO

Test case 3

Input:

abcdef
abc

Output:

YES

Tags: `string`, `startsWith`, `prefix`

CodeForge - B16 - Kiểm Tra Kết Thúc Bằng

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có kết thúc bằng hậu tố cho trước không (sử dụng `endsWith()`).

◇ Input

- Dòng 1: Chuỗi chính
- Dòng 2: Hậu tố cần kiểm tra

◇ Output

- In ra **YES** nếu chuỗi kết thúc bằng hậu tố
- In ra **NO** nếu không

◇ Constraints

- $1 \leq \text{độ dài chuỗi chính} \leq 10^5$
- $1 \leq \text{độ dài hậu tố} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
World
```

Output:

```
YES
```

Test case 2

Input:

```
Java Programming
Java
```

Output:

NO

Test case 3

Input:

document.pdf
.pdf

Output:

YES

Tags: `string`, `endsWith`, `suffix`

CodeForge - B17 - Đếm Số Từ

Độ khó: ★ ★ Medium

Đề bài

Đếm số từ trong chuỗi (các từ ngăn cách bởi khoảng trắng).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra số từ

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
2
```

Test case 2

Input:

```
Java Programming Language
```

Output:

```
3
```

Test case 3

Input:

```
Hello  World
```

Output:

```
2
```

Test case 4

Input:

```
0
```

Tags: `string`, `word-count`, `trim`

CodeForge - B18 - Đếm Nguyên Âm

Độ khó: ★ ★ Medium

Đề bài

Đếm số lượng nguyên âm (a, e, i, o, u) trong chuỗi (không phân biệt hoa thường).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra số lượng nguyên âm

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
3
```

Giải thích: e, o, o

Test case 2

Input:

```
AEIOU
```

Output:

5

Test case 3

Input:

bcdfg

Output:

0

Tags: `string`, `vowel-count`, `charAt`

CodeForge - B19 - Đảo Ngược Chuỗi

Độ khó: ★ ★ Medium

Đề bài

Đảo ngược chuỗi (không dùng StringBuilder/StringBuffer).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra chuỗi sau khi đảo ngược

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello
```

Output:

```
olleH
```

Test case 2

Input:

```
Java Programming
```

Output:

```
gnimmargorP avaJ
```

Test case 3

Input:

12345

Output:

54321

Tags: `string`, `reverse`, `charAt`

CodeForge - B20 - Kiểm Tra Palindrome

Độ khó: ★ ★ Medium

Đề bài

Kiểm tra chuỗi có phải là palindrome không (đọc xuôi ngược như nhau, không phân biệt hoa thường).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra **YES** nếu là palindrome
- In ra **NO** nếu không phải

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
racecar
```

Output:

```
YES
```

Test case 2

Input:

```
Hello
```

Output:

```
NO
```

Test case 3

Input:

Madam

Output:

YES

Test case 4

Input:

A man a plan a canal Panama

Output:

NO

Giải thích: Có khoảng trắng

Tags: `string`, `palindrome`, `two-pointers`

CodeForge - B21 - Đếm Ký Tự Xuất Hiện

Độ khó: ★ ★ Medium

Đề bài

Đếm số lần xuất hiện của một ký tự trong chuỗi (phân biệt hoa thường).

◇ Input

- Dòng 1: Chuỗi
- Dòng 2: Ký tự cần đếm

◇ Output

- In ra số lần xuất hiện

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
1
```

Output:

```
3
```

Test case 2

Input:

```
Java Programming
a
```

Output:

```
3
```

Test case 3

Input:

```
abcdef  
x
```

Output:

```
0
```

Tags: `string`, `frequency`, `charAt`

CodeForge - B22 - Viết Hoa Chữ Cái Đầu Mỗi Từ

Độ khó: ★ ★ Medium

Đề bài

Viết hoa chữ cái đầu tiên của mỗi từ trong chuỗi (Title Case).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra chuỗi sau khi viết hoa chữ cái đầu mỗi từ

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
hello world
```

Output:

```
Hello World
```

Test case 2

Input:

```
java programming language
```

Output:

```
Java Programming Language
```

Test case 3

Input:

```
HELLO WORLD
```

Output:

```
Hello World
```

Tags: `string`, `title-case`, `charAt`, `substring`

CodeForge - B23 - Xóa Tất Cả Khoảng Trắng

Độ khó: ★ ★ Medium

Đề bài

Xóa tất cả khoảng trắng trong chuỗi (không dùng replace).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra chuỗi sau khi xóa tất cả khoảng trắng

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
HelloWorld
```

Test case 2

Input:

```
Java Programming Language
```

Output:

```
JavaProgrammingLanguage
```

Test case 3

Input:

```
a b c
```

Output:

```
abc
```

Tags: `string`, `whitespace-removal`, `charAt`

CodeForge - B24A - Tìm Từ Dài Nhất

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Tìm từ dài nhất trong chuỗi. Nếu có nhiều từ cùng độ dài, in ra từ xuất hiện đầu tiên.

◇ Input

- Một dòng chứa chuỗi (các từ ngăn cách bởi khoảng trắng)

◇ Output

- Dòng 1: Từ dài nhất
- Dòng 2: Độ dài của từ đó

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World Java
```

Output:

```
Hello
5
```

Test case 2

Input:

```
Programming is fun
```

Output:

```
Programming  
11
```

Test case 3

Input:

```
a bb ccc bb
```

Output:

```
ccc  
3
```

Tags: `string`, `word-processing`, `max-length`

CodeForge - B25A - Đếm Tần Suất Ký Tự

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Đếm tần suất xuất hiện của mỗi ký tự trong chuỗi (phân biệt hoa thường).

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra mỗi ký tự và số lần xuất hiện, mỗi cặp trên một dòng
- Sắp xếp theo thứ tự xuất hiện đầu tiên

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
hello
```

Output:

```
h: 1
e: 1
l: 2
o: 1
```

Test case 2

Input:

```
aabbcc
```

Output:

```
a: 2  
b: 2  
c: 2
```

Test case 3

Input:

```
Java
```

Output:

```
J: 1  
a: 2  
v: 1
```

Tags: `string`, `frequency`, `counting`

CodeForge - B26A - Kiểm Tra Anagram

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Kiểm tra hai chuỗi có phải là anagram của nhau không (chứa cùng các ký tự với cùng tần suất, không phân biệt hoa thường).

◇ Input

- Dòng 1: Chuỗi thứ nhất
- Dòng 2: Chuỗi thứ hai

◇ Output

- In ra **YES** nếu là anagram
- In ra **NO** nếu không phải

◇ Constraints

- $1 \leq \text{độ dài mỗi chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
listen  
silent
```

Output:

```
YES
```

Test case 2

Input:

```
Hello  
World
```

Output:

NO

Test case 3

Input:

Dormitory
Dirty room

Output:

NO

Giải thích: Có khoảng trắng khác nhau

Test case 4

Input:

Astronomer
Moon starrer

Output:

NO

Tags: `string`, `anagram`, `frequency`, `sorting`

CodeForge - B27A - Tìm Chuỗi Con Dài Nhất Không Lặp

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Tìm độ dài của chuỗi con dài nhất không chứa ký tự lặp lại.

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra độ dài của chuỗi con dài nhất không có ký tự lặp

◇ Constraints

- $0 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
abcabcbb
```

Output:

```
3
```

Giải thích: "abc"

Test case 2

Input:

```
bbbbbb
```

Output:

1

Giải thích: "b"

Test case 3

Input:

pwwkew

Output:

3

Giải thích: "wke"

Test case 4

Input:

abcdef

Output:

6

Tags: `string`, `sliding-window`, `substring`, `unique`

CodeForge - B28A - Nén Chuỗi

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Nén chuỗi theo định dạng: ký tự + số lần xuất hiện liên tiếp.

Ví dụ: "aaabbc" → "a3b2c1"

Nếu chuỗi nén dài hơn chuỗi gốc, trả về chuỗi gốc.

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra chuỗi sau khi nén

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
aaabbc
```

Output:

```
a3b2c1
```

Test case 2

Input:

```
aabbcc
```

Output:

```
a2b2c2
```

Test case 3

Input:

```
abcd
```

Output:

```
abcd
```

Giải thích: Chuỗi nén "a1b1c1d1" dài hơn gốc

Test case 4

Input:

```
aaaaaaaaaa
```

Output:

```
a10
```

Tags: `string`, `compression`, `run-length-encoding`

CodeForge - B29A - Kiểm Tra Chuỗi Đối Xứng (Bỏ Qua Non-Alphanumeric)

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Kiểm tra chuỗi có phải palindrome không, chỉ xét các ký tự chữ và số, bỏ qua ký tự đặc biệt và khoảng trắng, không phân biệt hoa thường.

◇ Input

- Một dòng chứa chuỗi

◇ Output

- In ra **YES** nếu là palindrome
- In ra **NO** nếu không phải

◇ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 10^5$

Ví dụ

Test case 1

Input:

```
A man, a plan, a canal: Panama
```

Output:

```
YES
```

Test case 2

Input:

```
race a car
```

Output:

NO

Test case 3

Input:

Was it a car or a cat I saw?

Output:

YES

Test case 4

Input:

Madam, I'm Adam

Output:

YES

Tags: `string`, `palindrome`, `alphanumeric`, `validation`

CodeForge - B30A - Tìm Tất Cả Vị Trí Xuất Hiện

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Tìm tất cả các vị trí xuất hiện của chuỗi con trong chuỗi chính.

◇ Input

- Dòng 1: Chuỗi chính
- Dòng 2: Chuỗi con cần tìm

◇ Output

- In ra các vị trí xuất hiện (0-indexed), cách nhau bởi dấu cách
- Nếu không tìm thấy, in ra **-1**

◇ Constraints

- $1 \leq \text{độ dài chuỗi chính} \leq 10^5$
- $1 \leq \text{độ dài chuỗi con} \leq 10^3$

Ví dụ

Test case 1

Input:

```
ababcbababa  
aba
```

Output:

```
0 5 7
```

Test case 2

Input:

```
Hello World  
o
```

Output:

```
4 7
```

Test case 3

Input:

```
abcdef  
xyz
```

Output:

```
-1
```

Test case 4

Input:

```
aaaaaa  
aa
```

Output:

```
0 1 2 3 4
```

Tags: `string`, `indexOf`, `search`, `all-occurrences`