

CodeForge - B01 - Kiểm Tra Khớp Hoàn Toàn (matches)

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có khớp hoàn toàn với pattern không (sử dụng `String.matches()`).

◊ Input

- Dòng 1: Chuỗi cần kiểm tra
- Dòng 2: Pattern (regex)

◊ Output

- In ra **MATCH** nếu khớp hoàn toàn
- In ra **NO MATCH** nếu không khớp

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 1000
- $1 \leq$ độ dài pattern ≤ 500

Ví dụ

Test case 1

Input:

```
abc  
abc
```

Output:

```
MATCH
```

Test case 2

Input:

```
abcd  
abc
```

Output:

```
NO MATCH
```

Test case 3

Input:

```
12345  
\d+
```

Output:

```
MATCH
```

Tags: [regex](#), [matches](#), [basic](#)

CodeForge - B02 - Kiểm Tra Chứa Pattern (find)

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có chứa pattern không (sử dụng **Pattern** và **Matcher.find()**).

◇ Input

- Dòng 1: Chuỗi cần kiểm tra
- Dòng 2: Pattern (regex)

◇ Output

- In ra **FOUND** nếu tìm thấy pattern trong chuỗi
- In ra **NOT FOUND** nếu không tìm thấy

◇ Constraints

- $1 \leq$ độ dài chuỗi ≤ 1000
- $1 \leq$ độ dài pattern ≤ 500

Ví dụ

Test case 1

Input:

```
Hello World  
World
```

Output:

```
FOUND
```

Test case 2

Input:

```
Java Programming  
\d+
```

Output:

```
NOT FOUND
```

Test case 3

Input:

```
abc123def  
\d+
```

Output:

```
FOUND
```

Tags: [regex](#), [pattern](#), [matcher](#), [find](#)

CodeForge - B03 - Đếm Số Lần Xuất Hiện Pattern

Độ khó: ★ ★ Medium

Đề bài

Đếm số lần pattern xuất hiện trong chuỗi (sử dụng `Matcher.find()` trong loop).

◇ Input

- Dòng 1: Chuỗi
- Dòng 2: Pattern (regex)

◇ Output

- In ra số lần xuất hiện

◇ Constraints

- $1 \leq$ độ dài chuỗi $\leq 10^5$
- $1 \leq$ độ dài pattern ≤ 500

Ví dụ

Test case 1

Input:

```
abc abc abc  
abc
```

Output:

```
3
```

Test case 2

Input:

```
a1b2c3d4  
\d
```

Output:

4

Test case 3

Input:

```
Hello World  
[aeiou]
```

Output:

3

Tags: regex, matcher, count, find

CodeForge - B04 - Kiểm Tra Chỉ Chữ Cái

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi chỉ chứa chữ cái (a-z, A-Z).

Pattern: `^[a-zA-Z]+$`

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra YES nếu chỉ chứa chữ cái
- In ra NO nếu không

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 1000

Ví dụ

Test case 1

Input:

```
HelloWorld
```

Output:

```
YES
```

Test case 2

Input:

```
Hello123
```

Output:

NO

Test case 3

Input:

Hello World

Output:

NO

Tags: regex, character-class, alphabetic

CodeForge - B05 - Kiểm Tra Chỉ Chữ Số

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi chỉ chứa chữ số (0-9).

Pattern: `^\d+$` hoặc `^[0-9]+$`

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra YES nếu chỉ chứa chữ số
- In ra NO nếu không

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 1000

Ví dụ

Test case 1

Input:

```
12345
```

Output:

```
YES
```

Test case 2

Input:

```
123a45
```

Output:

NO

Test case 3

Input:

123.45

Output:

NO

Tags: regex, digit, numeric

CodeForge - B06 - Kiểm Tra Alphanumeric

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi chỉ chứa chữ cái và chữ số.

Pattern: `^[a-zA-Z0-9]+$` hoặc `^\w+$` (nếu không cho phép underscore thì dùng cái đầu)

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra YES nếu chỉ chứa chữ cái và số
- In ra NO nếu không

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 1000

Ví dụ

Test case 1

Input:

```
Hello123
```

Output:

```
YES
```

Test case 2

Input:

```
Hello_123
```

Output:

NO

Test case 3

Input:

Java2024

Output:

YES

Tags: regex, alphanumeric, character-class

CodeForge - B07 - Validate Email Đơn Giản

Độ khó: ★★ Medium

📝 Đề bài

Validate email format đơn giản: `username@domain.extension`

Pattern: `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`

◊ Input

- Một dòng chứa email

◊ Output

- In ra `VALID` nếu đúng format
- In ra `INVALID` nếu sai

◊ Constraints

- `1 ≤ độ dài email ≤ 100`

💻 Ví dụ

Test case 1

Input:

```
user@example.com
```

Output:

```
VALID
```

Test case 2

Input:

```
invalid.email
```

Output:

INVALID

Test case 3

Input:

user@domain.co.uk

Output:

VALID

Test case 4

Input:

@example.com

Output:

INVALID

Tags: regex, email, validation

CodeForge - B08 - Validate Số Điện Thoại

Độ khó: ★★ Medium

Đề bài

Validate số điện thoại Việt Nam: 10 chữ số, bắt đầu bằng 0.

Pattern: `^0\d{9}$`

◊ Input

- Một dòng chứa số điện thoại

◊ Output

- In ra `VALID` nếu đúng format
- In ra `INVALID` nếu sai

◊ Constraints

- `1 ≤ độ dài chuỗi ≤ 20`

Ví dụ

Test case 1

Input:

```
0123456789
```

Output:

```
VALID
```

Test case 2

Input:

```
1234567890
```

Output:

INVALID

Test case 3

Input:

012345678

Output:

INVALID

Tags: regex, phone, validation

CodeForge - B09 - Trích Xuất Tất Cả Số

Độ khó: ★★ Medium

📝 Đề bài

Trích xuất tất cả các số trong chuỗi (sử dụng `Matcher.group()`).

Pattern: \d+

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra tất cả các số tìm được, mỗi số trên một dòng
- Nếu không tìm thấy số nào, in ra NO NUMBERS

◊ Constraints

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

📊 Ví dụ

Test case 1

Input:

```
abc123def456ghi789
```

Output:

```
123  
456  
789
```

Test case 2

Input:

```
Price: $99 and $149
```

Output:

```
99  
149
```

Test case 3**Input:**

```
Hello World
```

Output:

```
NO NUMBERS
```

Tags: [regex](#), [extraction](#), [group](#), [digit](#)

CodeForge - B10 - Thay Thế Tất Cả Số Bằng X

Độ khó: ★ Easy

Đề bài

Thay thế tất cả các chữ số bằng 'X' (sử dụng `replaceAll()`).

Pattern: `\d`

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra chuỗi sau khi thay thế

◊ Constraints

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

Ví dụ

Test case 1

Input:

```
abc123def456
```

Output:

```
abcXXXdefXXX
```

Test case 2

Input:

```
Phone: 0123456789
```

Output:

Phone: XXXXXXXXXX

Test case 3

Input:

Hello World

Output:

Hello World

Tags: regex, replaceAll, substitution

CodeForge - B11 - Xóa Tất Cả Khoảng Trắng

Độ khó: ★ Easy

Đề bài

Xóa tất cả khoảng trắng trong chuỗi (sử dụng `replaceAll()`).

Pattern: `\s+`

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra chuỗi sau khi xóa khoảng trắng

◊ Constraints

- $1 \leq$ độ dài chuỗi $\leq 10^5$

Ví dụ

Test case 1

Input:

```
Hello World Java
```

Output:

```
HelloWorldJava
```

Test case 2

Input:

```
a b c
```

Output:

```
abc
```

Test case 3

Input:

```
NoSpaceHere
```

Output:

```
NoSpaceHere
```

Tags: `regex`, `replaceAll`, `whitespace`

CodeForge - B12 - Tách Chuỗi Theo Nhiều Dấu Phân Cách

Độ khó: ★★ Medium

Đề bài

Tách chuỗi theo nhiều dấu phân cách: dấu phẩy, chấm phẩy, khoảng trắng.

Pattern: [, ; \s]+

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra các phần tử sau khi tách, mỗi phần tử trên một dòng

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 10^5

Ví dụ

Test case 1

Input:

```
apple,banana;orange
```

Output:

```
apple
banana
orange
```

Test case 2

Input:

```
one two,three;four
```

Output:

```
one
two
three
four
```

Test case 3

Input:

```
a, b ; c d
```

Output:

```
a
b
c
d
```

Tags: `regex`, `split`, `multiple-delimiters`

CodeForge - B13 - Kiểm Tra Username Hợp Lệ

Độ khó: ★★ Medium

📝 Đề bài

Validate username:

- Độ dài 3-16 ký tự
- Chỉ chứa chữ cái, số và underscore
- Bắt đầu bằng chữ cái

Pattern: `^[a-zA-Z][a-zA-Z0-9_]{2,15}$`

◊ Input

- Một dòng chứa username

◊ Output

- In ra `VALID` nếu hợp lệ
- In ra `INVALID` nếu không hợp lệ

◊ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 50$

📊 Ví dụ

Test case 1

Input:

```
user123
```

Output:

```
VALID
```

Test case 2

Input:

```
123user
```

Output:

INVALID

Test case 3

Input:

ab

Output:

INVALID

Test case 4

Input:

valid_user_123

Output:

VALID

Tags: [regex](#), [username](#), [validation](#)

CodeForge - B14 - Validate Password

Độ khó: ★ ★ ★ Hard

Đề bài

Validate password mạnh:

- Ít nhất 8 ký tự
- Có ít nhất 1 chữ hoa
- Có ít nhất 1 chữ thường
- Có ít nhất 1 chữ số

◊ Input

- Một dòng chứa password

◊ Output

- In ra **STRONG** nếu đạt yêu cầu
- In ra **WEAK** nếu không đạt

◊ Constraints

- $1 \leq \text{độ dài password} \leq 100$

Ví dụ

Test case 1

Input:

```
Pass1234
```

Output:

```
STRONG
```

Test case 2

Input:

```
password
```

Output:

WEAK

Test case 3

Input:

PASSWORD123

Output:

WEAK

Test case 4

Input:

Abc123

Output:

WEAK

Giải thích: Chưa đủ 8 ký tự**Tags:** regex, password, validation, lookahead

CodeForge - B15 - Kiểm Tra Ngày Tháng (dd/mm/yyyy)

Độ khó: ★ ★ Medium

Đề bài

Kiểm tra format ngày tháng: dd/mm/yyyy (chỉ kiểm tra format, không validate logic).

Pattern: `^\d{2}/\d{2}/\d{4}$`

◊ Input

- Một dòng chứa ngày tháng

◊ Output

- In ra `VALID FORMAT` nếu đúng format
- In ra `INVALID FORMAT` nếu sai

◊ Constraints

- `1 ≤ độ dài chuỗi ≤ 50`

Ví dụ

Test case 1

Input:

```
25/12/2024
```

Output:

```
VALID FORMAT
```

Test case 2

Input:

```
1/1/2024
```

Output:

INVALID FORMAT

Test case 3

Input:

99/99/9999

Output:

VALID FORMAT

Giải thích: Đúng format nhưng không hợp lệ về logic

Tags: `regex`, `date`, `format-validation`

CodeForge - B16 - Trích Xuất URL

Độ khó: ★ ★ ★ Hard

Đề bài

Trích xuất tất cả các URL trong văn bản.

Pattern đơn giản: `https?:\/\/[^s]+`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả URL tìm được, mỗi URL trên một dòng
- Nếu không tìm thấy, in ra `NO URLs`

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

Ví dụ

Test case 1

Input:

```
Visit https://example.com and http://test.org
```

Output:

```
https://example.com
http://test.org
```

Test case 2

Input:

```
Check out https://github.com/user/repo for more info
```

Output:

```
https://github.com/user/repo
```

Test case 3

Input:

```
No links here
```

Output:

```
NO URLs
```

Tags: `regex`, `url`, `extraction`

CodeForge - B17 - Validate IPv4 Address

Độ khó: ★ ★ ★ Hard

Đề bài

Validate địa chỉ IPv4: 4 nhóm số (0-255) cách nhau bởi dấu chấm.

Pattern: `^((25[0-5]|2[0-4][0-9]|01)?[0-9]?)(\.){3}(25[0-5]|2[0-4][0-9]|01)?[0-9]?)$`

◊ Input

- Một dòng chứa địa chỉ IP

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 50

Ví dụ

Test case 1

Input:

```
192.168.1.1
```

Output:

```
VALID
```

Test case 2

Input:

```
256.1.1.1
```

Output:

INVALID

Test case 3

Input:

192.168.1

Output:

INVALID

Test case 4

Input:

0.0.0.0

Output:

VALID

Tags: regex, ipv4, validation, complex

CodeForge - B18 - Đếm Nguyên Âm

Độ khó: ★ Easy

Đề bài

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

Pattern: [aeiouAEIOU]

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra số lượng nguyên âm

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 10^5

Ví dụ

Test case 1

Input:

```
Hello World
```

Output:

```
3
```

Test case 2

Input:

```
AEIOU
```

Output:

5

Test case 3

Input:

bcdfg

Output:

0

Tags: regex, vowel, count

CodeForge - B19 - Xóa HTML Tags

Độ khó: ★ ★ Medium

Đề bài

Xóa tất cả HTML tags trong chuỗi.

Pattern: <[^>]+>

◊ Input

- Một dòng chứa HTML

◊ Output

- In ra văn bản sau khi xóa tags

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 10^5

Ví dụ

Test case 1

Input:

```
<p>Hello <b>World</b></p>
```

Output:

```
Hello World
```

Test case 2

Input:

```
<div><h1>Title</h1><p>Content</p></div>
```

Output:

TitleContent

Test case 3

Input:

Plain text

Output:

Plain text

Tags: regex, html, strip-tags

CodeForge - B20 - Validate Hex Color Code

Độ khó: ★ ★ Medium

Đề bài

Validate mã màu Hex: # theo sau bởi 3 hoặc 6 ký tự hex (0-9, a-f, A-F).

Pattern: `^#[0-9a-fA-F]{3}|[0-9a-fA-F]{6}$`

◊ Input

- Một dòng chứa mã màu

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 20

Ví dụ

Test case 1

Input:

```
#FFF
```

Output:

```
VALID
```

Test case 2

Input:

```
#FFFFFF
```

Output:

VALID

Test case 3

Input:

```
#GGG
```

Output:

INVALID

Test case 4

Input:

```
#12345
```

Output:

INVALID

Tags: [regex](#), [hex](#), [color](#), [validation](#)

CodeForge - B21 - Tìm Từ Lặp Lại

Độ khó: ★ ★ ★ Hard

Đề bài

Tìm các từ xuất hiện liên tiếp 2 lần (case insensitive).

Pattern: `\b(\w+)\s+\1\b` (sử dụng backreference)

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra các từ lặp lại, mỗi từ trên một dòng
- Nếu không có, in ra **NO DUPLICATES**

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

Ví dụ

Test case 1

Input:

```
This is is a test test
```

Output:

```
is
test
```

Test case 2

Input:

```
The the cat sat sat
```

Output:

```
the  
sat
```

Test case 3

Input:

```
No duplicates here
```

Output:

```
NO DUPLICATES
```

Tags: regex, backreference, duplicate, group

CodeForge - B22 - Chuẩn Hóa Khoảng Trắng

Độ khó: ★ ★ Medium

Đề bài

Chuẩn hóa khoảng trắng: thay thế nhiều khoảng trắng liên tiếp bằng 1 khoảng trắng, trim đầu cuối.

Pattern: \s+

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra chuỗi đã chuẩn hóa

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 10^5

Ví dụ

Test case 1

Input:

```
Hello      World      Java
```

Output:

```
Hello World Java
```

Test case 2

Input:

```
Good      Morning
```

Output:

Good Morning

Test case 3

Input:

Already normalized

Output:

Already normalized

Tags: regex, whitespace, normalization

CodeForge - B23 - Validate Credit Card (Simplified)

Độ khó: ★★ Medium

Đề bài

Validate số thẻ tín dụng đơn giản: 16 chữ số, có thể có dấu gạch ngang hoặc khoảng trắng mỗi 4 số.

Pattern: `^\d{4}[-\s]?\d{4}[-\s]?\d{4}[-\s]?\d{4}$`

◊ Input

- Một dòng chứa số thẻ

◊ Output

- In ra **VALID FORMAT** nếu đúng format
- In ra **INVALID FORMAT** nếu sai

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 30

Ví dụ

Test case 1

Input:

```
1234567890123456
```

Output:

```
VALID FORMAT
```

Test case 2

Input:

```
1234-5678-9012-3456
```

Output:

VALID FORMAT

Test case 3

Input:

```
1234 5678 9012 3456
```

Output:

VALID FORMAT

Test case 4

Input:

```
1234-567890123456
```

Output:

INVALID FORMAT

Tags: regex, credit-card, validation

CodeForge - B24 - Trích Xuất Email Addresses

Độ khó: ★ ★ ★ Hard

Đề bài

Trích xuất tất cả email addresses trong văn bản.

Pattern: `[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả email tìm được, mỗi email trên một dòng
- Nếu không tìm thấy, in ra `NO EMAILS`

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

Ví dụ

Test case 1

Input:

```
Contact: john@example.com or jane@test.org
```

Output:

```
john@example.com
jane@test.org
```

Test case 2

Input:

```
Email me at support@company.co.uk
```

Output:

```
support@company.co.uk
```

Test case 3

Input:

```
No emails here
```

Output:

```
NO EMAILS
```

Tags: `regex`, `email`, `extraction`

CodeForge - B25 - Validate Time Format (HH:MM)

Độ khó: ★★ Medium

📝 Đề bài

Validate thời gian format 24h: HH:MM (00:00 - 23:59).

Pattern: `^([01]\d|2[0-3]):([0-5]\d)$`

◊ Input

- Một dòng chứa thời gian

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 10

📊 Ví dụ

Test case 1

Input:

```
14:30
```

Output:

```
VALID
```

Test case 2

Input:

```
25:00
```

Output:

INVALID

Test case 3

Input:

00:00

Output:

VALID

Test case 4

Input:

23:59

Output:

VALID

Tags: regex, time, validation

CodeForge - B26 - Kiểm Tra Chuỗi Bắt Đầu Bằng Nguyên Âm

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có bắt đầu bằng nguyên âm không (không phân biệt hoa thường).

Pattern: `^[aeiouAEIOU]`

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra YES nếu bắt đầu bằng nguyên âm
- In ra NO nếu không

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 1000

Ví dụ

Test case 1

Input:

```
Apple
```

Output:

```
YES
```

Test case 2

Input:

```
Banana
```

Output:

NO

Test case 3

Input:

orange

Output:

YES

Tags: [regex](#), [vowel](#), [anchor](#)

CodeForge - B27 - Kiểm Tra Chuỗi Kết Thúc Bằng Số

Độ khó: ★ Easy

Đề bài

Kiểm tra chuỗi có kết thúc bằng chữ số không.

Pattern: \d\$

◊ Input

- Một dòng chứa chuỗi

◊ Output

- In ra YES nếu kết thúc bằng số
- In ra NO nếu không

◊ Constraints

- $1 \leq \text{độ dài chuỗi} \leq 1000$

Ví dụ

Test case 1

Input:

```
test123
```

Output:

```
YES
```

Test case 2

Input:

```
test
```

Output:

NO

Test case 3

Input:

version2

Output:

YES

Tags: regex, digit, anchor

CodeForge - B28 - Word Boundary

Độ khó: ★ ★ Medium

Đề bài

Tìm từ "cat" độc lập (không phải substring của từ khác) sử dụng word boundary.

Pattern: `\bcat\b`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra số lần từ "cat" xuất hiện độc lập

◊ Constraints

- 1 ≤ độ dài văn bản ≤ 10^5

Ví dụ

Test case 1

Input:

```
The cat is a cat
```

Output:

```
2
```

Test case 2

Input:

```
category concatenate
```

Output:

0

Test case 3

Input:

I have a cat. My cat is cute.

Output:

2

Tags: regex, word-boundary, whole-word

CodeForge - B29 - Thay Thế Số Điện Thoại

Độ khó: ★★ Medium

Đề bài

Ẩn số điện thoại: thay thế 6 số giữa bằng "***".

Ví dụ: 0123456789 → 012***6789

◊ Input

- Một dòng chứa số điện thoại (10 chữ số, bắt đầu bằng 0)

◊ Output

- In ra số điện thoại đã ẩn

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 20

Ví dụ

Test case 1

Input:

```
0123456789
```

Output:

```
012***6789
```

Test case 2

Input:

```
0987654321
```

Output:

```
098***4321
```

Test case 3

Input:

```
0111111111
```

Output:

```
011***1111
```

Tags: regex, masking, privacy

CodeForge - B30 - Validate MAC Address

Độ khó: ★★ Medium

Đề bài

Validate địa chỉ MAC: 6 nhóm 2 ký tự hex, cách nhau bởi dấu : hoặc -.

Pattern: `^([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$`

◊ Input

- Một dòng chứa địa chỉ MAC

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- $1 \leq$ độ dài chuỗi ≤ 30

Ví dụ

Test case 1

Input:

```
00:1A:2B:3C:4D:5E
```

Output:

```
VALID
```

Test case 2

Input:

```
00-1A-2B-3C-4D-5E
```

Output:

VALID

Test case 3

Input:

00:1A:2B:3C:4D

Output:

INVALID

Tags: regex, mac-address, validation

CodeForge - B31 - Tách Tên Họ

Độ khó: ★★ Medium

Đề bài

Tách họ tên thành các phần: Họ, Tên đệm (nếu có), Tên.

Giả sử format: "Họ TênĐệm Tên" hoặc "Họ Tên"

◊ Input

- Một dòng chứa họ tên đầy đủ

◊ Output

- Dòng 1: Họ
- Dòng 2: Tên đệm (nếu có, nếu không có in "N/A")
- Dòng 3: Tên

◊ Constraints

- $1 \leq$ độ dài họ tên ≤ 100

Ví dụ

Test case 1

Input:

```
Nguyen Van An
```

Output:

```
Nguyen  
Van  
An
```

Test case 2

Input:

```
Tran Minh
```

Output:

```
Tran  
N/A  
Minh
```

Test case 3

Input:

```
Le Thi Thu Ha
```

Output:

```
Le  
Thi Thu  
Ha
```

Tags: `regex`, `name-parsing`, `split`

CodeForge - B32 - Đếm Từ Viết Hoa

Độ khó: ★ ★ Medium

Đề bài

Đếm số từ bắt đầu bằng chữ cái viết hoa.

Pattern: `\b[A-Z][a-z]*`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra số từ viết hoa

◊ Constraints

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

Ví dụ

Test case 1

Input:

```
Hello World from Java
```

Output:

```
4
```

Test case 2

Input:

```
this is all lowercase
```

Output:

```
0
```

Test case 3

Input:

```
The Quick Brown Fox
```

Output:

```
4
```

Tags: regex, capitalized, count

CodeForge - B33 - Validate File Extension

Độ khó: ★★ Medium

📝 Đề bài

Kiểm tra file có extension hợp lệ không: jpg, .png, .gif, .pdf, .txt

Pattern: ^.+\.jpg|png|gif|pdf|txt\$

◊ Input

- Một dòng chứa tên file

◊ Output

- In ra VALID nếu extension hợp lệ
- In ra INVALID nếu không hợp lệ

◊ Constraints

- 1 ≤ độ dài tên file ≤ 100

💻 Ví dụ

Test case 1

Input:

```
document.pdf
```

Output:

```
VALID
```

Test case 2

Input:

```
image.jpg
```

Output:

VALID

Test case 3

Input:

file.exe

Output:

INVALID

Test case 4

Input:

noextension

Output:

INVALID

Tags: regex, file-extension, validation

CodeForge - B34A - Extract Hashtags

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Trích xuất tất cả hashtags từ văn bản (bắt đầu bằng # theo sau bởi chữ cái/số).

Pattern: `#[a-zA-Z0-9_]+`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả hashtags, mỗi hashtag trên một dòng
- Nếu không tìm thấy, in ra **NO HASHTAGS**

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

Ví dụ

Test case 1

Input:

```
Learning #Java and #Python today! #coding
```

Output:

```
#Java  
#Python  
#coding
```

Test case 2

Input:

```
Check out #2024trends and #AI_revolution
```

Output:

```
#2024trends  
#AI_revolution
```

Test case 3**Input:**

```
No hashtags here
```

Output:

```
NO HASHTAGS
```

Tags: [regex](#), [hashtag](#), [extraction](#), [social-media](#)

CodeForge - B35A - Validate URL Complete

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Validate URL đầy đủ hơn: protocol, domain, optional port, path, query.

Pattern cơ bản: `^https?://[a-zA-Z0-9.-]+\. [a-zA-Z]{2,}(:\d+)?(/.*)?$`

◊ Input

- Một dòng chứa URL

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- `1 ≤ độ dài URL ≤ 1000`

Ví dụ

Test case 1

Input:

```
https://www.example.com
```

Output:

```
VALID
```

Test case 2

Input:

```
http://localhost:8080/api/users
```

Output:

VALID

Test case 3

Input:

```
ftp://example.com
```

Output:

INVALID

Test case 4

Input:

```
https://api.github.com/users?page=1
```

Output:

VALID

Tags: `regex`, `url`, `validation`, `complex`

CodeForge - B36A - Parse Log Entry

Độ khó: ★ ★ ★ Hard (Advanced)

📝 Đề bài

Parse log entry format: [LEVEL] timestamp - message

Pattern: `^[\(INFO|WARN|ERROR)\]\s+(\S+)\s+-\s+(.+)$`

Trích xuất: level, timestamp, message

◊ Input

- Một dòng chứa log entry

◊ Output

- Dòng 1: Level
- Dòng 2: Timestamp
- Dòng 3: Message
- Nếu không match, in ra INVALID LOG FORMAT

◊ Constraints

- 1 ≤ độ dài log ≤ 1000

📊 Ví dụ

Test case 1

Input:

```
[INFO] 2024-01-15T10:30:00 - Application started
```

Output:

```
INFO  
2024-01-15T10:30:00  
Application started
```

Test case 2

Input:

```
[ERROR] 2024-01-15T10:35:00 - Database connection failed
```

Output:

```
ERROR  
2024-01-15T10:35:00  
Database connection failed
```

Test case 3

Input:

```
Invalid log entry
```

Output:

```
INVALID LOG FORMAT
```

Tags: [regex](#), [log-parsing](#), [extraction](#), [groups](#)

CodeForge - B37A - Validate Strong Password Advanced

Độ khó: ★★☆ Hard (Advanced)

Đề bài

Validate password siêu mạnh:

- Ít nhất 12 ký tự
- Có ít nhất 1 chữ hoa
- Có ít nhất 1 chữ thường
- Có ít nhất 1 chữ số
- Có ít nhất 1 ký tự đặc biệt (@#\$%^&+=)

Sử dụng positive lookahead.

◊ Input

- Một dòng chứa password

◊ Output

- In ra **VERY STRONG** nếu đạt yêu cầu
- In ra **WEAK** nếu không đạt

◊ Constraints

- $1 \leq \text{độ dài password} \leq 100$

Ví dụ

Test case 1

Input:

```
MyP@ssw0rd123
```

Output:

```
VERY STRONG
```

Test case 2

Input:

```
Password123
```

Output:

```
WEAK
```

Test case 3**Input:**

```
Abc@123
```

Output:

```
WEAK
```

Giải thích: Chưa đủ 12 ký tự

Tags: regex, password, validation, lookahead, complex

CodeForge - B38A - Extract Mentions (Social Media)

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Trích xuất tất cả mentions (bắt đầu bằng @ theo sau bởi username).

Pattern: `@[a-zA-Z0-9_]{3,15}`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả mentions, mỗi mention trên một dòng
- Nếu không tìm thấy, in ra **NO MENTIONS**

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

Ví dụ

Test case 1

Input:

```
Thanks @john_doe and @jane_smith for the help!
```

Output:

```
@john_doe  
@jane_smith
```

Test case 2

Input:

```
Follow @java_dev for coding tips
```

Output:

```
@java_dev
```

Test case 3

Input:

```
No mentions here
```

Output:

```
NO MENTIONS
```

Tags: regex, mention, extraction, social-media

CodeForge - B39A - Validate Date Advanced (dd-MM-yyyy)

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Validate ngày tháng với logic cơ bản:

- dd: 01-31
- MM: 01-12
- yyyy: 1900-2099

Pattern: `^(0[1-9]|1[0-2])-(0[1-9]|1[0-2])-(19|20)\d{2}$`

◊ Input

- Một dòng chứa ngày tháng

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- `1 ≤ độ dài chuỗi ≤ 20`

Ví dụ

Test case 1

Input:

```
25-12-2024
```

Output:

```
VALID
```

Test case 2

Input:

32-01-2024

Output:

INVALID

Test case 3**Input:**

15-13-2024

Output:

INVALID

Test case 4**Input:**

01-01-1899

Output:

INVALID

Tags: regex, date, validation, advanced

CodeForge - B40A - Remove Comments (Programming)

Độ khó: ★★☆ Hard (Advanced)

Đề bài

Xóa comments trong code:

- Single-line comments: // comment
- Multi-line comments: /* comment */

Pattern cho single-line: `//.*`

◊ Input

- Một dòng chứa code

◊ Output

- In ra code sau khi xóa comments

◊ Constraints

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

Ví dụ

Test case 1

Input:

```
int x = 5; // This is a comment
```

Output:

```
int x = 5;
```

Test case 2

Input:

```
String name = "John"; // User name
```

Output:

```
String name = "John";
```

Test case 3

Input:

```
System.out.println("Hello");
```

Output:

```
System.out.println("Hello");
```

Tags: regex, comment, code-processing

CodeForge - B41A - Extract Numbers From Mixed Text

Độ khó: ★ ★ ★ Hard (Advanced)

📝 Đề bài

Trích xuất tất cả số (bao gồm số thực) từ văn bản.

Pattern: `-?\d+\.\?\d*`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả số tìm được, mỗi số trên một dòng
- Nếu không tìm thấy, in ra **NO NUMBERS**

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

📊 Ví dụ

Test case 1

Input:

```
Price: $99.99 and $149.50
```

Output:

```
99.99  
149.50
```

Test case 2

Input:

```
Temperature: -5.5 degrees, max 25.3
```

Output:

```
-5.5  
25.3
```

Test case 3

Input:

```
No numbers here
```

Output:

```
NO NUMBERS
```

Tags: regex, number, extraction, decimal

CodeForge - B42A - Camel Case to Snake Case

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Chuyển đổi camelCase sang snake_case.

Ví dụ: "myVariableName" → "my_variable_name"

Sử dụng regex để tìm chữ hoa và thay thế.

◊ Input

- Một dòng chứa chuỗi camelCase

◊ Output

- In ra chuỗi snake_case

◊ Constraints

- 1 ≤ độ dài chuỗi ≤ 1000

Ví dụ

Test case 1

Input:

```
myVariableName
```

Output:

```
my_variable_name
```

Test case 2

Input:

```
getUserData
```

Output:

```
get_user_data
```

Test case 3

Input:

```
HTTPResponse
```

Output:

```
h_t_t_p_response
```

Tags: regex, case-conversion, naming-convention

CodeForge - B43A - Validate JSON Key Format

Độ khó: ★ ★ ★ Hard (Advanced)

📝 Đề bài

Validate JSON key format: phải là chuỗi trong dấu ngoặc kép, chỉ chứa chữ, số, underscore.

Pattern: `^"[a-zA-Z_][a-zA-Z0-9_]*"$`

◊ Input

- Một dòng chứa key

◊ Output

- In ra **VALID** nếu hợp lệ
- In ra **INVALID** nếu không hợp lệ

◊ Constraints

- `1 ≤ độ dài key ≤ 100`

💻 Ví dụ

Test case 1

Input:

```
"user_name"
```

Output:

```
VALID
```

Test case 2

Input:

```
"firstName"
```

Output:

VALID

Test case 3

Input:

"123invalid"

Output:

INVALID

Test case 4

Input:

user_name

Output:

INVALID

Tags: regex, json, validation, key-format

CodeForge - B44A - Extract Quoted Strings

Độ khó: ★ ★ ★ Hard (Advanced)

📝 Đề bài

Trích xuất tất cả chuỗi trong dấu ngoặc kép.

Pattern: `"([^\"]*)"`

Sử dụng capturing group để lấy nội dung bên trong.

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả chuỗi trong dấu ngoặc kép (không bao gồm dấu ngoặc), mỗi chuỗi trên một dòng
- Nếu không tìm thấy, in ra **NO QUOTED STRINGS**

◊ Constraints

- $1 \leq$ độ dài văn bản $\leq 10^5$

📊 Ví dụ

Test case 1

Input:

```
He said "Hello" and "Goodbye"
```

Output:

```
Hello  
Goodbye
```

Test case 2

Input:

```
The message is "Welcome to Java"
```

Output:

```
Welcome to Java
```

Test case 3

Input:

```
No quotes here
```

Output:

```
NO QUOTED STRINGS
```

Tags: regex, quoted-string, extraction, capturing-group

CodeForge - B45A - Advanced Phone Number Extraction

Độ khó: ★★☆ Hard (Advanced)

Đề bài

Trích xuất số điện thoại với nhiều format:

- 0123456789
- 012-345-6789
- (012) 345-6789
- +84 123 456 789

Pattern phức tạp: `(\+\d{1,3}\s?)?(\(?(\d{3})\)?[\s.-]?)?\d{3}[\s.-]?\d{4}`

◊ Input

- Một dòng chứa văn bản

◊ Output

- In ra tất cả số điện thoại tìm được, mỗi số trên một dòng
- Nếu không tìm thấy, in ra `NO PHONE NUMBERS`

◊ Constraints

- `1 ≤ độ dài văn bản ≤ 10^5`

Ví dụ

Test case 1

Input:

```
Call me at 0123456789 or (012) 345-6789
```

Output:

```
0123456789  
(012) 345-6789
```

Test case 2

Input:

```
International: +84 123 456 789
```

Output:

```
+84 123 456 789
```

Test case 3

Input:

```
No phone numbers here
```

Output:

```
NO PHONE NUMBERS
```

Tags: [regex](#), [phone](#), [extraction](#), [complex](#), [multiple-formats](#)