Hello, **2024101067**.

UNIX Path Simplification

Submit solution

My submissions
All submissions
Best submissions

✓ **Points:** 100 (partial)

Time limit: 1.0s

Memory limit: 256M

Allowed languages
C

Problem Statement

You are given an absolute Unix-style path. Your task is to simplify the path by resolving special tokens:

- A single dot (`"."`) represents the current directory and should be ignored.
- A double dot (`".."`) represents moving up one directory. If you are at the root directory, it should be ignored.
- Multiple consecutive slashes should be treated as a single slash.

The simplified canonical path must always begin with a slash (`"/"`) and must be the shortest string representing the absolute path.

Input

The input consists of a single line containing an absolute Unix-style path:

- The path is a non-empty string.
- It always starts with a slash (`"/"`).
- The length of the string is at most 1,000,000 characters.(Suggested to use fgets(buff, n, stream);)

Output

Print the simplified canonical path.

Examples

proudly powered by **DMOJ** | English (en)

Copy

```
/home/
```

Output:

Copy

```
/home
```

Explanation: The trailing slash is removed, so the canonical path is simply `/home`.

Example 2

Input:

Copy

```
/../
```

Output:

Copy

```
/
```

Explanation: The `".."` token attempts to move up from the root, which is not allowed. Therefore, the simplified path remains `/`.

Example 3

Input:

Copy

```
/home//foo/
```

Output:

Copy

```
/home/foo
```

Explanation: Multiple consecutive slashes are replaced by a single slash.

Example 4

Input:

```
/a/./b/../../c/
```

Output:

Copy

```
/c
```

❓ Clarifications

Request clarification

No clarifications have been made at this time.