Problem 3	**The Path Less Traversed**	6 Points

Program Name: path.cpp Input File: path.dat

Introduction

The following is a brief recap of Unix pathnames:

- UNIX pathnames use the forward slash (/) as a directory separator.
- The root directory has a pathname of "/".
- "." refers to "this directory." It is generic across the directory structure, so whatever directory you're in can be referred to as ".". "." is pronounced "dot."
- Similarly, ".." refers to the directory immediately above the current directory. This directory is also called the parent directory. Again, this reference is generic across the file system, so ".." refers to the parent directory of your current directory, wherever that may be. ".." is pronounced "dot-dot."

Input

Input to this problem will consist of a (non-empty) series of up to 100 data sets. Each data set will be formatted according to the following description, and there will be **no blank lines** separating data sets.

A single data set has 1 component:

1. *Start line* - A single line, "A B", where:

A will be a path name representing a directory tree. A will start with a '/' (referred to as the "root" directory), followed by directory names delimited by a single '/'. A directory is considered a subdirectory of the directory immediately preceding it in the list (conversely, the directory preceding a directory in the list is considered its parent directory). For example, if A is "/subdir1/subdir2", the directories are "/", "subdir1", and "subdir2", where "subdir1" is a subdirectory of "/", and "subdir2" is a subdirectory of "subdir1", "subdir1" is the parent directory of "subdir2", and "/" is the parent directory of "subdir1". Directory names will consist of alphanumeric characters only. The maximum length of A will be 100 characters.

B will be an absolute path name to be traversed. B will always start with a '/' followed by a series of '.', '..', and directory names, delimited by a single '/'. For example, B could be "/./subdir1/../subdir1". The maximum length of B will be 100 characters.

Output

For each data set, there will be exactly one line of output. If B is a valid path name, the output will consist of the absolute path name of the directory reached after traversing B for the directory tree represented by A. In the example above, the output would be "/subdir1". Traversal is accomplished reading B left to right as follows:

A starting '/' in B (which is always the case) means you start in the root directory.

A '.' in B means you traverse to the directory you are currently in.

A '..' in B means you traverse to the parent directory of the directory you are currently in, unless you are in the root directory, in which case you traverse to the root directory.

A directory name in B means you traverse to that directory, if it is a subdirectory of the directory you are currently in. Otherwise, B is not a valid path name.

If B is not a valid path name, the output will consist of a single line with the statement "INVALID DIRECTORY".

Example: Input File

```
/subdir1/subdir2 /./subdir1/../subdir1
/subdir1/subdir2/subdir3 /././././/subdir1/subdir2/subdir3/../../subdir2
/subdir1/subdir2/subdir3 /././././/subdir1/subdir2/subdir3/../../subdir3
```

Output to screen

```
/subdir1
/subdir1/subdir2
INVALID DIRECTORY
```