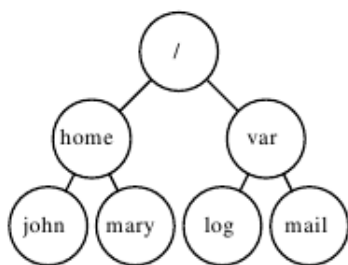


5957 Relative Paths

Most users of modern (and many not-so-modern) operating systems are familiar with the concept of a **path**, which is a string giving the route in a filesystem tree from a given starting node (a directory) to a desired destination node. An **absolute path** starts at the root of the tree and always begins with “/”. A **relative path** gives similar information, but starts from the **current directory** (also called the **current working directory**), which is just a specified node in the tree; a relative path never starts with “/”. The names of nodes in the path are separated by “/” (Apologies to Windows® users.). A node name of “.” refers to the current node, and a node name of “..” refers to the parent of the current node (one level higher in the tree). By convention, the node “/” is its own parent, so “/..” and “/.” are the same as “/”.

Examples: Consider a few simple examples based on the simple filesystem tree shown to the left.



- “/home/john” is an absolute path to the lower-left node in the tree, and
- “/var/mail” is an absolute path to the lower-right node in the tree.
- “/home/mary/..” is an absolute path equivalent to “/home”.

If the current directory is “/var”, then the relative path “../home/./mary” is equivalent to the absolute path “/home/mary”. Finally, if the current directory is “/var/mail”, then the relative path “../../home/john/.” is equivalent to the absolute path “/home/john”.

Given a path P that is either absolute or relative, and an absolute path C to the current directory, determine the relative path S to the same node identified by P such that S is the shortest possible correct relative path (that is, has the fewest characters). Note that every path must have at least one character.

Input

There will be multiple cases to consider, each of which will contain two lines of input. The first line will contain the path P , and the second line will contain an absolute path C . Node names will consist only of between 1 and 8 lowercase alphabetic characters, but “/”, “.” and “..” may also appear in paths. No input line will contain whitespace (tab or space characters). Input and output paths will never be longer than 72 characters.

Output

For each input case, display five lines, as follows. On the first line, display the input case number (1, 2, ...). On the next line display ‘P = ’ (indented by three spaces) and the value of P . On the next two lines, display C and S in a similar manner. The last line of output for each case should be a blank line. This format is illustrated in the samples.

Sample Input

```
a/b
/c/d
/
/a/b/c
/a/b
/a/c
/
/
/x/y
/x/y
/a/b/..c
/a/d
/a/b
/c/d
/c/a/b
/c/d
/c/d/a/b
/c/d
```

Sample Output

Case 1:

```
P = a/b
C = /c/d
S = a/b
```

Case 2:

```
P = /
C = /a/b/c
S = ../.../..
```

Case 3:

```
P = /a/b
C = /a/c
S = ../b
```

Case 4:

```
P = /
C = /
S = .
```

Case 5:

```
P = /x/y
C = /x/y
S = .
```

Case 6:

```
P = /a/b/..c
C = /a/d
```

S = ../c

Case 7:

P = /a/b

C = /c/d

S = ../../a/b

Case 8:

P = /c/a/b

C = /c/d

S = ../a/b

Case 9:

P = /c/d/a/b

C = /c/d

S = a/b