

题目描述:

实现一个模拟目录管理功能的软件，输入一个命令序列，输出最后一条命令运行结果。

支持命令:

- 1) 创建目录命令: `mkdir 目录名称`, 如 `mkdir abc` 为在当前目录创建 `abc` 目录, 如果已存在同名目录则不执行任何操作。此命令无输出。
- 2) 进入目录命令: `cd 目录名称`, 如 `cd abc` 为进入 `abc` 目录, 特别地, `cd ..` 为返回上级目录, 如果目录不存在则不执行任何操作。此命令无输出。
- 3) 查看当前所在路径命令: `pwd`, 输出当前路径字符串。

约束:

- 1) 目录名称仅支持小写字母; `mkdir` 和 `cd` 命令的参数仅支持单个目录, 如: `mkdir abc` 和 `cd abc`; 不支持嵌套路径和绝对路径, 如 `mkdir abc/efg`, `cd abc/efg`, `mkdir /abc/efg`, `cd /abc/efg` 是不支持的。
- 2) 目录符号为 `/`, 根目录 `/` 作为初始目录。
- 3) 任何不符合上述定义的有效命令不做任何处理并且无输出。

输入描述:

输入 N 行字符串, 每一行字符串是一条命令。

输出描述:

输出最后一条命令运行结果字符串。

补充说明:

命令行数限制 100 行以内, 目录名称限制 10 个字符以内。

示例 1

输入:

```
mkdir abc
cd abc
pwd
```

输出:

```
/abc/
```

说明:

在根目录创建一个 `abc` 的目录并进入 `abc` 目录中查看当前目录路径, 输出当前路径 `/abc/`。

利用字典和栈模拟

```
root = {}
ans = ''
cur = root
stack = []
```

```
while True:
```

```
    try:
```

```
        line = input().split(' ')
```

```
        ans = ''
```

```

    if len(line) == 2:
        if line[0] == 'mkdir':
            if line[1].isalpha() and line[1].lower() == line[1] and line[1] not
in cur:
                cur[line[1]] = {'..': cur}
            elif line[0] == 'cd':
                if line[1] in cur:
                    cur = cur[line[1]]
                    if line[1] == '..':
                        stack.pop()
                    else:
                        stack.append(line[1])
            elif len(line) == 1 and line[0] == 'pwd':
                if stack:
                    ans = '/' + '/'.join(stack) + '/'
                else:
                    ans = '/'
except:
    if ans:
        print(ans)
    break

```