

Problema F

Pastas

Arquivo fonte: `pastas.{c | cpp | java}`

Autor: *Leandro Luque (Fatec Mogi das Cruzes)*

Méridjeine é uma programadora *front-end* experiente. Ela é fascinada por tecnologias como Javascript, jQuery, Ionic, Angular, entre outras. É até difícil conversar com a menina. No Whatsapp, ao invés de falar tchau, ela manda um `'window.close()'`. E por aí vai.

Em seu mais recente projeto, um disco virtual que irá fazer frente ao mega.nz (um dos únicos nos quais seus arquivos estão teoricamente seguros), ela precisou implementar uma árvore Javascript que representa a estrutura de pastas do disco. Ela sabe que existem várias bibliotecas Javascript prontas que fazem isso, mas ela não gosta de códigos de terceiros. Segundo ela, código bom é o que ela escreve. A árvore pode ter qualquer quantidade de níveis. A Figura 1 ilustra uma árvore com 3 níveis.

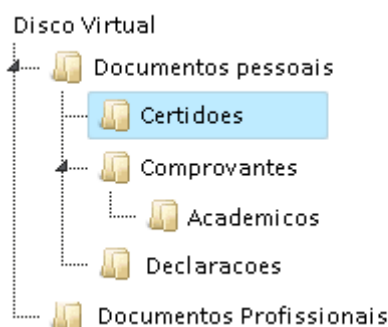


Figura 1. Exemplo de árvore de pastas com 3 níveis.

No presente momento do projeto, ela precisa implementar uma funcionalidade que pesquisa pastas por parte do nome. Para cada pasta encontrada, ela deseja imprimir o caminho completo da pasta. Para isso, ela pediu a sua ajuda.

Entrada

A entrada é composta por vários casos de teste. A primeira linha de cada caso de teste contém um número inteiro N ($1 \leq N \leq 200$), que indica o número de pastas que serão informadas. As próximas N linhas contêm um número inteiro I ($1 \leq I \leq 200$), um texto de tamanho máximo T ($1 \leq T \leq 50$) e um número inteiro P ($0 \leq P \leq 200$), separados por espaços em branco. O número I é um identificador único da pasta, o texto T é o nome da pasta e o número P é o identificador da pasta pai – isto é, o identificador da pasta dentro da qual a pasta especificada nesta linha se encontra. Um valor 0 (zero) indica a pasta raiz. Caso a pasta especificada na linha esteja localizada na raiz da árvore, o identificador P será igual a 0 (zero). A próxima linha do caso de teste contém um inteiro M ($1 \leq M \leq 10$), que indica o número de pesquisas que serão realizadas. Por fim, as próximas M linhas contêm um texto S cada (o tamanho máximo de S é T) com um texto que deverá ser pesquisado na árvore.

Saída

Para cada pesquisa de cada caso de teste, imprima o caminho completo da pasta encontrada ou o texto “NOT FOUND” (sem aspas) se nenhuma pasta com o nome especificado tiver sido encontrada. Caso mais de uma pasta seja encontrada em uma pesquisa, imprima todas as pastas (uma por linha), assumindo a ordem alfabética. O texto digitado na pesquisa (entrada) não precisa ser exatamente igual ao nome da pasta. Basta que o nome da pasta contenha este texto para que ela seja

retornada. Desta forma, se o texto procurado for 'Documento', tanto a pasta 'Documentos Pessoais' quanto 'Documentos Profissionais' será um resultado válido da pesquisa. A pesquisa é *case-sensitive*.

Exemplos

Entrada

```
6
1 Documentos Pessoais 0
3 Certidoes 1
2 Documentos Profissionais 0
5 Academicos 4
6 Declaracoes 1
4 Comprovantes 1
5
Declara
Root
Doc
C
O
```

Saída

```
\Documentos Pessoais\Declaracoes
NOT FOUND
\Documentos Pessoais
\Documentos Profissionais
\Documentos Pessoais\Certidoes
\Documentos Pessoais\Comprovantes
\Documentos Pessoais
\Documentos Pessoais\Comprovantes
\Documentos Pessoais\Comprovantes\Academicos
\Documentos Profissionais
```