

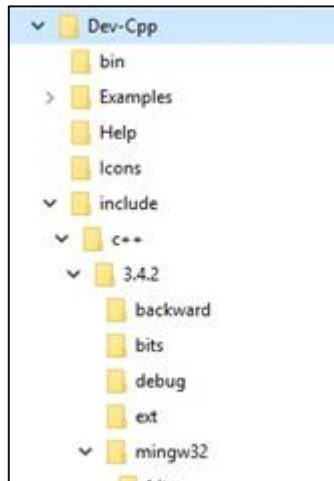
Atividade de Programação 9

Diretório de Arquivos

Algoritmos e Estruturas de Dados I

Tempo limite: 1s

Em Sistemas Operacionais a disposição de arquivos e pastas é realizada de forma hierárquica. Uma forma de representar essa estrutura é utilizando uma árvore genérica.



As seguintes operações podem ser realizadas na estrutura de diretórios, a partir da entrada de dados em programa que simula o funcionamento da estrutura:

-a <nome do novo elemento (arquivo ou pasta) > <nó pai (pasta)> => ADICIONAR

-m <nome do elemento (arquivo) > <nó pai (pasta)> => MOVER

-r <nome do elemento (arquivo)> => REMOVER

Tanto os nomes de arquivos como os nomes de diretórios são compostos dos caracteres minúsculos dos intervalos [a-z] (sem acentuação ou caracteres especiais) e [0-9], como, por exemplo, *file23* (arquivo), *\pasta02t* (pasta). As pastas iniciam sempre com o caractere “\”. Por exemplo, *\dir*, *\programa*, etc.

Deve-se escrever um programa que construa uma árvore genérica para representar a estrutura de arquivos e pastas, e realizar as operações de adicionar, remover e mover elementos no diretório. Em seguida deve-se exibir na tela o caminho de pastas/diretórios do elemento buscado, caso exista (por simplificação não haverá elementos com nome repetidos na árvore final).

Entrada

A entrada para cada teste se inicia informando o número de operações N ($1 \leq N \leq 1024$), depois deve-se informar o nome do arquivo (*string*) de tamanho S ($1 \leq S \leq 1024$) que será buscada. Nas próximas N linhas devem ser fornecidas as operações, conforme descrito anteriormente. Os nomes dos diretórios e arquivos também têm tamanhos iguais à S . Percebe-se nos exemplos abaixo que o diretório `\root` não é adicionado, logo deve-se criá-lo antes da execução de qualquer operação. O número de máximo de elementos da árvore é E , onde $1 \leq E \leq 1024$.

Saída

A saída do programa é exibida em uma única linha, exibindo o nome do arquivo procurado e o caminho em que este arquivo está alocado, caso exista. A *string*, que representa esse caminho, é composta pelos nomes das pastas/diretórios que compõem o caminho da busca, seguindo a ordem do nó folha (arquivo encontrado) até o nó raiz (`\root`).

Caso o arquivo não seja encontrado deve-se exibir na saída o texto:

Arquivo nao encontrado!

Dicas:

- Pode-se utilizar funções da biblioteca `<string.h>` para manipular *strings*;
- A operação de mover é basicamente uma remoção e uma adição (inserção) do arquivo;
- A remoção é feita **sempre** nas folhas (**arquivos**), nunca em pastas. Nota-se que os arquivos são sempre folhas da árvore.
- A operação de mover ocorre também somente em arquivos.

Restrições:

- O programa deve ser escrito em C/C++;
- Deve-se ter no código as funções de: inicialização, inserção, remoção, busca e movimentação para a árvore genérica;
- Deve-se utilizar uma TAD Árvore Genérica, conforme mostrado em sala:
<https://classroom.google.com/c/NjE3Njg0Mzg2NDIw/m/NjQwODQwNjI1NjM2/details>

Exemplo de Entrada	Exemplo de Saída
--------------------	------------------

5 file1 -a \dir1 \root -a \dir2 \dir1 -a file1 \dir2 -m file1 \dir1 -a file2 \dir2	file1 \dir1 \root
Exemplo de Entrada	Exemplo de Saída
7 aula2 -a \dir1 \root -a \dir2 \root -a aula1 \dir1 -a aula2 \dir1 -a aula3 \dir1 -r aula2 -r aula1	Arquivo nao encontrado!
Exemplo de Entrada	Exemplo de Saída
15 arq5 -a \dir1 \root -a \dir2 \root -a arq1 \dir1 -a arq4 \dir2 -m arq4 \dir1 -m arq1 \dir2 -a arq2 \dir1 -a arq5 \dir2 -m arq5 \dir1 -m arq2 \dir2 -a arq3 \dir1 -a arq6 \dir2 -m arq6 \dir1 -m arq3 \dir2 -r arq2	arq5 \dir1 \root