

Tarefa 7 - O mensageiro

Prazo de entrega: 15/11/2020

[Voltar para lista de tarefas](#)

Um mensageiro tem um conjunto de *tokens*, que nada mais é do que um saco de pequenos cartõezinhos numerados e com algum texto no verso. Os número dos cartões são únicos, ou seja, não há dois cartões com o mesmo número. Ao serem “reorganizados”, os textos nos versos dos cartões formam uma mensagem.

Esse mensageiro tem o dever de transmitir a mensagem ao rei, mas como a mensagem está cifrada, ele deve passar por uma sequência de autoridades, com hierarquia cada vez mais alta. Cada autoridade tem um número pessoal privado, que só ela conhece. Para cada autoridade, o mensageiro deve perguntar o seu número e procurar uma **tríade** de cartões cuja a soma dos valores seja igual ao número dito. O mensageiro deve então remover esses três cartões do saco e entregá-los à autoridade, que devolverá **um novo cartão** com seu número e um novo texto no verso.

Depois de ter passado por todas as autoridades, o mensageiro vai até o rei. Nesse momento, ele lê a mensagem obtida ao juntar **todos** os cartões restantes de forma com que os números fiquem em ordem crescente.

A sua tarefa é escrever um programa em C que recebe um conjunto de cartões do mensageiro e uma sequência de inteiros com os números das autoridades em **ordem de visita**. O programa deve devolver a mensagem a ser transmitida ao rei. Suponha que o texto contido em um cartão devolvido por uma autoridade é a concatenação dos textos contidos na tríade entregue pelo mensageiro e que a concatenação é realizada seguindo a ordem numérica de cada cartão.

Entrada

A entrada é composta de diversas mensagens. Para cada mensagem, são dadas três linhas: A primeira contém dois inteiros *m* e *n*, que representam o número de cartões e o número de autoridades, respectivamente. A segunda linha contém *m* pares, onde o primeiro item é o número do cartão e o segundo item é texto do verso entre aspas duplas e de no máximo 5 caracteres. A terceira linha contém *n* inteiros, onde cada inteiro é o número autoridade. A entrada termina com final de arquivo.

Exemplo de entrada

```
3 1
6 "reci" 3 "p" 7 "oso"
16
4 1
1 "Meu " 5 "pr" 7 "so!" 6 "ecio"
18
5 2
1 "Vou" 2 "ma " 4 "via" 8 " nu" 16 "gem."
22 31
```

Saída

A saída do seu programa deverá apresentar uma linha para cada mensagem.

Exemplo de saída

```
precioso
Meu precioso!
Vou numa viagem.
```

Correção

Esta tarefa será corrigida automaticamente e por um monitor. O teste automático será realizado sempre que você realizar um `push`. Para indicar que a sua tarefa está pronta para a correção pelo monitor, remova o arquivo `nao_corrigir.txt` do seu repositório antes de dar `commit`.

A nota dependerá do número de casos de teste privado executados corretamente. No entanto, nota poderá ser descontada por questões de critérios específicos, organização, legibilidade ou eficiência. Pense na próxima pessoa que lerá seu código, pode ser você daqui a um mês!

Critérios específicos

É **obrigatório** implementar um algoritmo que utiliza a estrutura de árvore binária de busca. Com exceção da estrutura árvore binária de busca e strings, **é vedado** o uso de qualquer outra estrutura de dados para armazena coleções de dados (vetores, listas, pilhas, etc.). É vedado também a utilização de qualquer algoritmo de ordenação.