

MC202 (Estruturas de Dados) - 1s2021

Tarefa 04: Recursão

Prof^o Ricardo Dahab

Assistente: Elisa Dell'Arriva

Assistente: Jônatas Trabuco Belotti

Instituto de Computação - UNICAMP

Sobre a Tarefa

O objetivo desta tarefa é que o(a) aluno(a) se familiarize com o conceito de recursão e a implementação de algoritmos recursivos. A tarefa consiste em calcular o preço final de um produto p .

Cada produto p é composto por um conjunto de n subprodutos $L_p = \{p_1, p_2, p_3, \dots, p_n\}$, e possui um percentual de imposto a ser aplicado I_p . Caso um produto não possua nenhum subproduto, ou seja, $n = 0$, então ele é considerado uma matéria-prima e, nesses casos, o produto possui apenas um preço base B_p e o percentual de imposto I_p .

O preço final de um produto p é calculado pela função recursiva $f(p)$, definida como:

$$f(p) = \begin{cases} B_p \times I_p & , \text{ se } n = 0 \\ \left(\sum_{p_i \in L_p} f(p_i) \right) \times I_p & , \text{ caso contrário.} \end{cases} \quad (1)$$

Instruções

Defina uma **struct** chamada **Produto** para armazenar todos os dados de cada produto. Mais detalhes sobre o armazenamento de um produto são dados posteriormente.

Para cada caso de teste, teremos uma sequência de produtos. Tal sequência deve ser armazenada em um vetor de ponteiros, em que cada ponteiro aponta para a **struct** (do tipo **Produto**) que representa o respectivo produto. Na Figura 1, mostramos uma esquematização de tal vetor.

O algoritmo que calcula o preço final de um produto deve obrigatoriamente ser uma função recursiva. Essa função deve receber um ponteiro para a **struct** do produto cujo preço final deseja-se calcular e o vetor com todos os produtos. A função deve retornar o preço final do produto calculado pela Equação 1.

Utilize o tipo **float** para armazenar números com casas decimais.

O não cumprimento de qualquer dessas instruções implicará em nota 0 (zero) nesta tarefa.

Formato da entrada e saída

A primeira linha da entrada é um inteiro n , entre 1 e 1000, que representa a quantidade de produtos a serem lidos. Cada uma das próximas n linhas contém o registro de um produto, composto por id, o número de subprodutos (entre 1 e 50), os ids de cada um dos k subprodutos (separados por um espaço) e o percentual de imposto aplicado sobre este produto (entre 0,0 e 100,0):

<id> <num_subprodutos> <id_subproduto_1> ... <id_subproduto_k> <imposto>

Quando o produto for uma matéria prima no lugar dos ids dos subprodutos é apresentado o preço base:

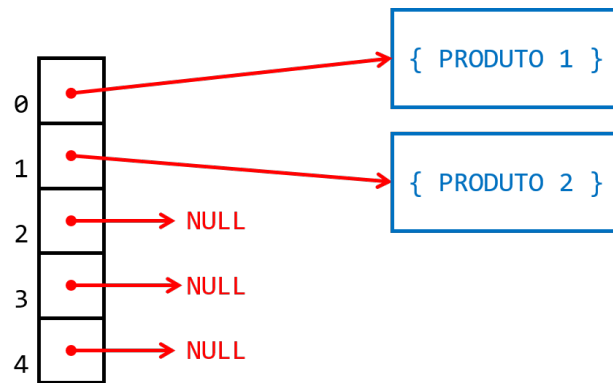


Figura 1: Vetor de ponteiros para as **structs** de cada produto.

```
<id> 0 <preço_base> <imposto>
```

A última linha da entrada contém o id do produto para o qual se deve calcular o preço final.

É esperada apenas 1 (uma) linha como saída, contendo o preço final do produto. Devem ser apresentadas apenas 2 (duas) casas decimais. O último caractere da saída deve obrigatoriamente ser um `\n`.

Exemplo 1.

Entrada:

```
3
1 2 2 3 17.8
2 0 23.1 7
3 0 12.98 9.3
1
```

Saída:

```
45.83
```

Exemplo 2.

Entrada:

```
4
12 3 15 20 26 31
15 0 23.1 8
20 0 12.99 15
26 0 43 12
15
```

Saída:

```
24.95
```

Exemplo 3.

Entrada:

```
5
6 0 13 7
1 0 22 9
12 2 1 6 14
35 2 1 12 11
8 4 6 1 12 35 22
8
```

Saída:

```
189.89
```

Dica. Note que a *struct* `Produto` deve conter um campo para armazenar o id do produto, um campo para armazenar o imposto do produto, um campo para dizer se o produto é uma matéria-prima ou não, um campo para o preço final do produto, um campo para armazenar o número de subprodutos que compõem o produto e um vetor dos ids de tais subprodutos.

Alguns avisos e lembretes

- A página da disciplina no SuSy é <https://susy.ic.unicamp.br:9999/mc202abc>.
- Para submeter, utilizem somente os dígitos numéricos do RA e a senha da DAC.
- O número máximo de submissões é 10.
- Esta tarefa tem peso 1 (um).
- A tarefa estará aberta até 10/05/2021, às 23h59.