

Aula Prática 5 - Estruturas de Dados I (BCC202)

Marco Antonio M. Carvalho
Universidade Federal de Ouro Preto
Departamento de Computação

1 de julho de 2021

Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.

1 Manipulação de listas encadeadas

Luiggy gosta de fazer amizades e sempre está expandindo sua lista de amigos na rede social *ListBook*. O *ListBook* permite que você inclua novos amigos em sua lista de amizade e os indique também para outros amigos da sua rede, tudo isso dinamicamente, sem limites de amigos na sua rede. Como Luiggy é seu amigo, ele pediu que você criasse um programa para facilitar a vida dele com esta tarefa. Para isso, Luiggy teve a seguinte ideia:

- O programa deverá ler a lista atual de amigos de Luiggy;
- O programa deverá ler uma nova lista de amigos de Luiggy;
- O programa deverá ler o nome de um amigo da lista atual que receberá a nova lista como indicação de amigos.

Especificação da Entrada

Você deve ler em uma única linha a lista de amigos de Luiggy, contendo somente o primeiro nome e separados por um espaço em branco. Na segunda linha é informada a nova lista de amigos, novamente, contendo somente o primeiro nome e separados por um espaço em branco. Na terceira e última linha, é informado o nome do amigo para o qual a nova lista de amigos será indicada. Caso não se queira indicar para ninguém a nova lista de amigos, a última linha conterá a palavra “*nao*” (sem acento e sempre em letras minúsculas).

Especificação da Saída

Seu programa deverá exibir a lista de amigos de Luiggy atualizada, ou seja, a união da lista original com a nova lista de amigos. Entretanto, se houver indicação para um amigo que já consta da lista original, os amigos da nova lista deverão ser inseridos **antes** do nome deste amigo. Caso não haja indicação, os novos nomes deverão ser inseridos no fim da lista de amigos original.

Exemplo de Entrada 1

```
Jones Pedro Carlos Lucas
Juca Valdineia Jovander
Carlos
```

Exemplo de Saída 1

```
Jones Pedro Juca Valdineia Jovander Carlos Lucas
```

Exemplo de Entrada 2

```
Jones Pedro Carlos Lucas
Juca Valdineia Jovander
nao
```

Exemplo de Saída 2

```
Jones Pedro Carlos Lucas Juca Valdineia Jovander
```

Estrutura do código

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Cada uma das listas de amigos deve ser armazenada em uma respectiva **lista encadeada** para cada caso de teste. A criação da lista de amigos final deve ser realizada com base na manipulação das duas listas encadeadas originais. As funções *TLista_Inicia*, *TLista_InserFinal*, *TLista_RetiraPrimeiro* e *TLista_Imprime* são as mesmas vistas na aula teórica com modificações mínimas para tratar strings. As funções *TLista_Esvazia* (remove cada elemento de uma lista e libera a memória), *TLista_append* (acrescenta o conteúdo de uma lista ao final de outra, apenas manipulando ponteiros) e *TLista_include* (inclui o conteúdo de uma lista em outra, a partir de uma posição específica, apenas manipulando ponteiros) devem ser implementadas pelo aluno. A função *TLista_EhVazia* foi adaptada do material original.

Diretivas de Compilação

```
$ gcc lista.c -c
$ gcc principal.c -c
$ gcc lista.o principal.o -o programa
```