MC202 - Estruturas de Dados

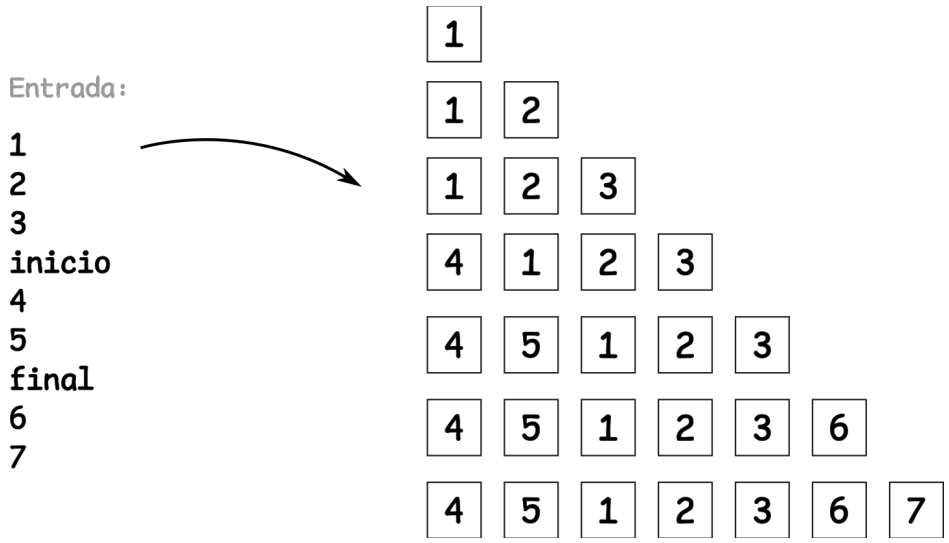
Lab 06

Data da Primeira Chance: 18 de Setembro de 2023
Link da atividade: https://classroom.github.com/a/2vbZfwYe
Peso: 3

Wickerbottom, A Bibliotecária, acaba de adquirir um exemplar do vigésimo sexto livro da série Duna: *The Heir of Caladan (2022),* mas ela se deu conta de que sua pilha de livros não lidos está definitivamente longa demais. Agora ela quer conferir todos os livros e decidir quais ela pretende ler primeiro e quais ela não faz tanta questão assim e vai deixar pro final.

A bibliotecária pensou em fazer um programa em C no Winona9000 (agora o computador tem nome) para criar sua lista de leitura, mas ela ainda não aprendeu a manipular listas ligadas. Ela sabe que usar vetores não seria uma boa ideia porque inserir no início é O(n), ou seja, muito custoso (ela já manja das análises assintóticas). Já que James, Winona, Jonathan, Elisa, a CAESB e o homem-morcego falaram muito bem do seu trabalho, ela agora pede para que você a ajude a implementar um programa em C que monta a lista na ordem correta enquanto ela busca a pilha de livros.

O plano é: ela vai digitar os títulos dos livros que devem ser inseridos no final na lista. Quando Wickerbottom decidir voltar para o início, ela digitará “inicio” e quando ela decidir adicionar no final ela digitará “final”. Note que quando ela volta para o início, os próximos livros devem seguir em ordem. Veja a figura abaixo.



Tem um último detalhe! À medida que ela vai adicionando os livros como entrada, pode ser que ela decida não ler mais algum dos livros que já foi inserido. Então, sempre que for digitado “remover <item>” você deve remover <item> da lista. Nesse caso, a inserção continua na “frente” do último livro adicionado, como seria normalmente. Se o livro removido for o último livro adicionado e a Wickerbottom não especificar “inicio” ou “final”, a próxima inserção deve ser feita no lugar do livro que foi removido (como se ele nunca tivesse sido adicionado).

Entrada

A entrada é composta por várias linhas de no máximo 55 caracteres. Cada linha pode conter uma das cinco opções:

- “adicionar <livro-tal>” para indicar que <livro-tal> deve ser adicionado. na frente do último que foi adicionado
- “inicio” para indicar que o próximo livro deve ser inserido no início da lista
- “final” para indicar que o próximo livro deve ser inserido no final da lista
- “remover <livro-tal>” para indicar que <livro-tal> deve ser removido da lista
- “Imprimir” para indicar que a lista deve ser impressa na tela

Note que um novo livro deve ser inserido depois do último que foi adicionado na lista, a não ser que a entrada anterior tenha sido “início”, “final” ou que a lista esteja vazia.

Não se preocupe, pois ela não tem livros repetidos, então nenhum livro será adicionado mais de uma vez.

A entrada encerra com fim de arquivo (EOF). Dica: você pode usar o retorno da função scanf para saber quantos itens foram lidos com sucesso e comparar esse retorno com a constante “EOF”.

Saída

A saída é composta por uma linha para cada “imprimir” dado como entrada. Cada linha contém todos os livros presentes na lista separados por uma vírgula e um espaço.

Exemplos

Exemplo 1:

Entrada

```
adicionar Lux Aeterna Redux

adicionar Lunar Grimoire

adicionar Tempering Temperatures

adicionar Overcoming Arachnophobia

imprimir
```

Saída

```
Lux Aeterna Redux, Lunar Grimoire, Tempering Temperatures,
Overcoming Arachnophobia
```

Exemplo 2:

Entrada

```
adicionar Boruto

imprimir

adicionar Horticulture Abridged

inicio

adicionar Applied Silviculture

adicionar Lux Aeterna

remover Boruto

imprimir
```

Saída

```
Boruto
Applied Silviculture, Lux Aeterna, Horticulture Abridged
```

Exemplo 3:

Entrada

```
adicionar Introduction to Algorithms

adicionar The C Programming Language

adicionar The Algorithm Design Manual
```

```
imprimir

inicio

adicionar Effective Java

remover Effective Java

adicionar Competitive Programming 3

imprimir
```

Saída

```
Introduction to Algorithms, The C Programming Language, The Algorithm Design Manual
Competitive Programming 3, Introduction to Algorithms, The C Programming Language, The Algorithm Design Manual
```

Regras e Avaliação

Seu código será avaliado não apenas pelos testes do GitHub, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código analisaremos neste laboratório:

- A escolha de bons nomes de funções e variáveis;
- A ausência de trechos de código repetidos desnecessariamente;
- O uso apropriado de funções;
- A ausência de vazamentos de memória;
- A eficiência dos algoritmos propostos;
- A correta utilização das Estruturas de Dados;
- **Deve ser implementada uma lista ligada.**

Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

Submissão

Você deverá submeter no repositório criado no aceite da tarefa. Você pode enviar arquivos adicionais caso deseje para serem incluídos por `livros.c`.
Não se esqueça de dar `git push` !

Lembre-se que sua atividade será corrigida automaticamente na aba “Actions” do repositório. Confirme a correção e o resultado, já que o que vale é o que está lá e não na sua máquina.

Atenção: O repositório da sua atividade conterá alguns arquivos iniciais. Fica **estritamente proibido** ao aluno alterar os arquivos já existentes, tais como o testador existente ou demais arquivos de configuração do laboratório.

Após a correção da primeira entrega, será aberta uma segunda chance, com prazo de entrega apropriado.