

MC202 - Estruturas de Dados

Lab 11 - Hashing

Link da atividade: <https://classroom.github.com/a/IJLX4BZp>

Peso: 5

É tempo de *Halloween*, e na cidade de Hashnópolis, no dia 31 de outubro, as crianças saem de porta em porta para o famoso “*Trick or treat*”. E com suas fantasias elaboradas, as famílias geralmente preferem “*Treat*” ao invés de “*Trick*”.



Como Hashnópolis é muito grande, é inviável que as crianças saiam batendo em todas as portas. Para ajudá-las, a prefeitura propôs um sistema em tempo real que auxiliará as crianças dizendo quais casas estão mais próximas de cada criança e quais casas possuem doces também.

Como é um sistema complexo, a prefeitura contratou você para desenvolver tal sistema. Considerando que Hashnópolis possui muitas casas, o sistema precisa ser eficiente, e eles têm certeza que você sabe o melhor jeito de implementar o que eles querem para o sistema.

O sistema precisa ser dinâmico, e ficará disponível para as famílias atualizarem, uma vez que os doces de uma casa podem acabar ou a família pode precisar sair de casa. Dessa forma, deve ser permitido que as famílias atualizem a quantidade de doces, e entrem ou saiam do sistema (para evitar problemas, é necessário que as famílias, quando não estão em casa, sejam excluídas do sistema).

Quando uma criança acessar o sistema, ela terá como retorno um conjunto de casas próximas, dessa forma, ela não precisa sair batendo em todas as portas de todas as casas, indo diretamente nas casas indicadas. O problema é que as casas são indicadas pelo sobrenome das famílias, então você vai precisar indexar as informações usando esses nomes. O sistema de recomendação de qual casa cada criança deve ir já está pronto, então você só precisa implementar o sistema para gerenciar os dados de cada família, assim o sistema de recomendação pode usá-los eficientemente.

Ao visitar cada família, cada criança recebe uma nota da família para sua fantasia, e essa nota define quantos doces cada criança vai ganhar. A fantasia pode ser boa, mediana, ou ruim, resultando em 3, 2 e 1 doces, respectivamente. Caso a família não tenha a quantidade suficiente de doces conforme a nota, a criança receberá o máximo de doces que a família puder dar.

Entrada

A entrada do programa é um conjunto de linhas que podem ser atualizações do sistema por uma família, ou uso do sistema por uma criança. Porém, todas começam com uma instrução. De acordo com cada instrução, o sistema deverá executar a ação em questão. As instruções podem ser:

- **ENTRAR:** Essa instrução representa a inclusão de uma família no sistema. Ela possui o formato:

ENTRAR <Nome Da Família> <Qtd de Doces>

Onde o nome da família é uma *string* de no máximo 10 caracteres sem espaços, e a quantidade de doces é um número inteiro. O número máximo de entradas de famílias é de 1500.

- **SAIR:** Representa a remoção de uma família do sistema. Possui o formato:

SAIR <Nome Da Família>

Onde o nome da família é uma *string* de no máximo 10 caracteres sem espaços.

- **COMPRAR:** Representa a compra de mais doces por uma família. Possui o formato:

COMPRAR <Nome Da Família> <Qtd de Doces>

Onde o nome da família é uma *string* de no máximo 10 caracteres sem espaços, e a quantidade de doces é um número inteiro, que é o número de doces comprados pela família.

- **TRICKORTREAT:** Representa o acesso de uma criança. Possui o formato:

TRICKORTREAT <Nome da Criança> <Sobrenome da Criança> <Qtd de Casas>

Onde o nome da criança é uma *string* de no máximo 10 caracteres (sem espaços) e o sobrenome da criança é uma *string* de no máximo 10 caracteres (sem espaços), e a

quantidade de casas é um número **N** inteiro entre 0 e 15, inclusive. Na sequência, existirão **N** linhas no formato:

<Nome da Família> <Nota>

Onde o nome da família é uma *string* de no máximo 10 caracteres sem espaços, e a nota pode ser “Boa”, “Media” ou “Ruim”, indicando a quantidade de doces que a criança vai receber daquela família.

- **FINALIZAR:** Representa o final do programa. Possui o formato:

FINALIZAR

Obs.: Todos os nomes de família e crianças são únicos e cada criança acessa o sistema apenas uma vez.

Saída

- Para cada criança, imprima o nome e a quantidade de doces que ela recebeu no total, seguindo o formato (sem aspas):
<Nome da Criança> recebeu <Qtd de Doces> doce(s) das famílias.
- Caso uma criança fique sem nenhum doce, ela receberá um incentivo da prefeitura, neste caso você deve imprimir:
<Nome da Criança> recebeu 10 doces da prefeitura.
- Cada vez que uma família ficar sem doces, imprima uma linha no formato:
A família <Nome da Família> ficou sem doces.
- Cada vez que uma família compra doces, imprima uma linha informando o novo total de doces no formato:
A família <Nome da Família> agora possui <Qtd de Doces> doces.
- Cada vez que uma família sai de casa, imprima uma linha informando o número de doces que ainda restavam naquela casa.
A família <Nome da Família> saiu com <Qtd de Doces> doce(s) sobrando.

Obs.: A ordem de impressão deve seguir os acontecimentos do programa.

Dicas

Existe um número máximo de famílias que o sistema suporta, procure um valor primo para definir o tamanho máximo da sua tabela hash conforme o número máximo de entradas.

Exemplos

Exemplo 1:

Entrada

```
ENTRAR Ricken 19
ENTRAR Wilson 25
ENTRAR Rocco 10
TRICKORTREAT Renata Silva 1
Ricken Boa
COMPRAR Ricken 25
TRICKORTREAT Arthur Costa 2
Ricken Media
Wilson Boa
SAIR Wilson
FINALIZAR
```

Saída

```
Renata Silva recebeu 3 doce(s) das familias.
A familia Ricken agora possui 41 doces.
Arthur Costa recebeu 5 doce(s) das familias.
A familia Wilson saiu com 22 doce(s) sobrando.
```

Exemplo 2:

Entrada

```
ENTRAR Ross 10
ENTRAR Silva 10
ENTRAR Costa 10
TRICKORTREAT Giovana Ester 3
Ross Boa
Silva Boa
Costa Boa
TRICKORTREAT Arthur Heinz 3
Ross Boa
```

```
Silva Media
Costa Ruim
TRICKORTREAT Yara Sven 3
Ross Boa
Silva Boa
Costa Media
COMPRAR Ross 10
TRICKORTREAT Leandro Hast 3
Ross Boa
Silva Media
Costa Media
TRICKORTREAT Pedro Const 1
Silva Boa
FINALIZAR
```

Saída

```
Giovana Ester recebeu 9 doce(s) das familias.
Arthur Heinz recebeu 6 doce(s) das familias.
Yara Sven recebeu 8 doce(s) das familias.
A familia Ross agora possui 11 doces.
A familia Silva ficou sem doces.
Leandro Hast recebeu 7 doce(s) das familias.
Pedro Const recebeu 10 doces da prefeitura.
```

Exemplo 3:

Entrada

```
ENTRAR Riven 10
ENTRAR Einstein 100
TRICKORTREAT Giovanna Silva 2
Riven Boa
Einstein Ruim
TRICKORTREAT Arthur Costa 1
```

```
Einstein Ruim
TRICKORTREAT Pedro Percival 0
FINALIZAR
```

Saída

```
Giovanna Silva recebeu 4 doce(s) das familias.
Arthur Costa recebeu 1 doce(s) das familias.
Pedro Percival recebeu 10 doces da prefeitura.
```

Regras e Avaliação

Para este laboratório, é necessário o uso da estrutura de dados do tipo Hash com **endereçamento aberto**. Operações de inclusão e remoção **deverão** ser implementadas.

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;
- Você deve implementar uma estrutura de hash com endereçamento aberto com inclusão e remoção de itens.

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 sua solução no repositório criado no aceite da tarefa. Você pode enviar arquivos adicionais caso deseje para serem incluídos por `halloween.c`.

Não se esqueça de dar `git push` !

Atenção: O repositório da sua atividade conterà 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.

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.