

# Trabalho 2 CI1057 - Tabelas Hash

Paulo Ricardo Lisboa de Almeida

1º Semestre - 2023

## 1 Descrição

Implemente uma Tabela Hash com hash estático em C. A Tabela Hash deve ter chaves inteiras, e não aceita duplicadas. Colisões devem ser resolvidas por encadeamento em uma lista duplamente encadeada.

## 2 Template

Os algoritmos devem ser implementados utilizando como base o template disponibilizado no Moodle.

O arquivo *tabelaHash.h* contém os protótipos das funções que obrigatoriamente devem ser implementadas, no arquivo *tabelaHash.c*.

Não é permitida a mudança nos protótipos de funções disponibilizados no arquivo *tabelaHash.h* (exceto onde explicitamente marcado no arquivo .h). É permitida a criação de funções auxiliares se necessário. Também é permitida a criação de arquivos de header (.h) e de implementação (.c) complementares se necessário.

Você deve alterar a struct definida no arquivo de template para que ela represente corretamente um nodo. A chave dos nodos deve ser um inteiro.

O programa deve compilar sem erros ou avisos através do comando *make*.

O nome do binário gerado deve ser *trab1\_GRR1* (e.g., *trab1\_1234*). Ajuste isso no *makefile*. Deve ser possível compilar e executar o binário nas máquinas do DInf.

## 3 Formato de Entrada e Saída

A entrada será feita pela entrada padrão (*stdin*). A entrada será uma sequência de linhas, onde cada linha representa uma operação. As entradas podem ser as seguintes:

- inserir um valor na tabela: i val
- remover um valor da tabela: r val
- buscar um valor na tabela: b val
- imprimir a tabela: l

- finalizar e sair: f

A exceção a essa regra é o primeiro item digitado, que sempre vai ser um número inteiro maior que zero, indicando o tamanho da Tabela Hash que deverá ser criada.

A saída será feita pela saída padrão (*stdout*). Saídas são geradas apenas quando o usuário solicitar uma impressão, ou buscar um item. Casos de erro também devem ser impressos (exemplo: remover um item que não existe). Veja um exemplo de execução do programa a seguir, onde os itens em *itálico* são entradas, e em **negrito** são saídas geradas pelo programa.

#### Trabalho de Nome do Aluno

Login Dinf seuLogin

GRR 1234

*10*

*i 0*

*i 1*

*i 2*

*l*

**0 [0] -> NULL**

**1 [1] -> NULL**

**2 [2] -> NULL**

**3 NULL**

**4 NULL**

**5 NULL**

**6 NULL**

**7 NULL**

**8 NULL**

**9 NULL**

*r 0*

*l*

**0 NULL**

**1 [1] -> NULL**

**2 [2] -> NULL**

**3 NULL**

**4 NULL**

**5 NULL**

**6 NULL**

**7 NULL**

**8 NULL**

**9 NULL**

*r 0*

**Falha ao remover 0.**

*i 1*

**Falha ao inserir 1.**

*i 0*

*i 3*

*i 4*

*i 10*

*i 110*

```

i 120
r 110
l
0 [0] -> [10] -> [120] -> NULL
1 [1] -> NULL
2 [2] -> NULL
3 [3] -> NULL
4 [4] -> NULL
5 NULL
6 NULL
7 NULL
8 NULL
9 NULL
f

```

A saída do seu programa deve seguir exatamente o modelo disposto neste documento, sob pena de descontos de nota ou até anulação do seu trabalho.

## 4 Função de Hash

A seguinte função de hash deve ser usada:

$$h(k) = k \bmod m$$

onde  $m$  é o tamanho da tabela hash. Cuidado, já que a chave  $k$  é um inteiro, e seu programa deve funcionar para chaves negativas. Você deve tratar esse problema ao calcular a função de hash.

## 5 Arquivos a serem entregues

Os arquivos devem estar em um diretório de nome *trab2\_GRR*. Se, por exemplo, o grr do aluno é 1234, o diretório contendo os arquivos deve se chamar *trab2\_1234*. Compacte esse diretório, sendo que a versão compactada vai se chamar *trab2\_1234.tar.gz*. Ao descompactar o arquivo *trab2\_1234.tar.gz*, deverá ser criado um diretório de nome *trab2\_1234* que conterá todos os demais arquivos. O diretório deve conter o seguinte:

- arquivos de código fonte .c e .h do programa;
- makefile;
- README.txt com no máximo 6.000 caracteres, contendo a documentação sintetizada do sistema implementado. Qualquer particularidade deve estar descrita neste texto (ex.: como foi implementada a função de hash?).

Não inclua quaisquer outros arquivos irrelevantes para o projeto. Por exemplo, não inclua binários compilados, ou arquivos objeto (.o). A inclusão de arquivos irrelevantes pode acarretar em descontos de nota.

## 6 Entrega

O trabalho deve ser entregue via Moodle. A data limite para o envio está estipulada no link de entrega. Não serão aceitas entregas em atraso, exceto para os casos explicitamente amparados pelas resoluções da UFPR.

## 7 Grupos, Pesos e Datas

**Grupos:** trabalho individual.

**Valor:** 15% da nota do semestre.

**Submissão:** Via Moodle. Veja a data limite no link de submissão do Moodle.

## 8 Distribuição da Nota

Alguns descontos padrão, considerando uma nota entre 0 e 100 pontos para o trabalho:

- Plágio: perda total da pontuação para todos os envolvidos. Isso é válido mesmo para casos onde o plágio se refere a apenas um trecho do código;
- Algoritmos implementados em linguagem diferente de C serão desconsiderados;
- Não submissão via Moodle acarreta na perda total dos pontos;
- Falta de algum arquivo requisitado: desconto de 10 a 100 pontos;
- Inclusão de arquivos desnecessários (lixo): desconto de 5 a 20 pontos;
- Erros e avisos de compilação: desconto de 5 a 100 pontos;
- Nomes de arquivo incorretos: 10 pontos por arquivo;
- Arquivo com formato incorreto: 10 a 100 pontos;
- Problemas de programação, como uso de funções descontinuadas, programação insegura, e outros problemas de design: 10 a 100 pontos;
- Corrupção, vazamentos de memória, *user after free*, *double free*, entre outros problemas de gerenciamento de memória: 10 a 100 pontos;
- Formato de saída incorreto: 10 a 100 pontos;
- Algoritmos implementados incompatíveis com uma Tabela Hash: 5 a 100 pontos;
- Para cada caso de teste (criado pelo professor, que serão diferentes dos cedidos aos alunos) que o programa falhar, serão descontados de 5 a 10 pontos.

## 9 Demais Regras

- Dúvidas ou casos não especificados neste documento podem ser discutidos com o professor até a **data de entrega do trabalho**. Não serão aceitas reclamações após a data da entrega.
- Os alunos podem (e devem) procurar o professor para tirar dúvidas quanto ao trabalho.
- O descumprimento das regras dispostas nesse documento podem acarretar na perda parcial ou total da nota do trabalho.