

1º Trabalho Prático: GBC034 - Algoritmos e Estruturas de Dados 2

Instruções

Número de integrantes por grupo 3 pessoas.
Grupos com mais ou menos pessoas somente serão aceitos se não houver número suficiente de alunos para formar outro grupo.

Data de Entrega:

07/05/2020

Forma de Entrega:

O trabalho deverá ser entregue por e-mail. No campo Assunto do e-mail, colocar:
Trabalho GBC034

No arquivo principal (**main**) do programa deve haver o seguinte comentário com os nomes dos integrantes do grupo:

```
/*  
Integrantes:  
Fulano de Tal - matrícula  
Ciclano de Tal - matrícula  
Beltrano de Tal - matrícula  
*/
```

Problema

Dada a implementação da **Tabela Hash** vista em aula, faça adaptações no código de modo a implementar uma **Tabela Hash** com **Endereçamento Aberto** e com **Redimensionamento dinâmico** (*Dynamic resizing*) para armazenar dados de alunos:

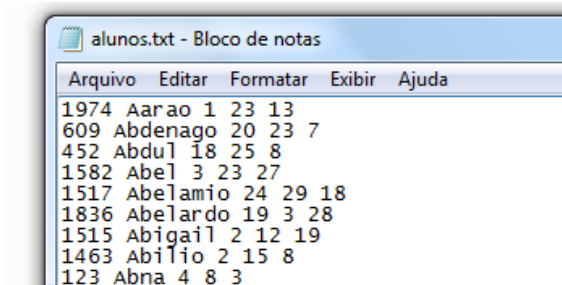
- Inicialmente, a **Tabela Hash** é definida com tamanho inicial **M**
- A cada inserção deve-se verificar quantas das **M** posições da **Tabela Hash** estão ocupadas. Caso mais de 75% das posições esteja ocupada (*load fator*), a tabela deverá ser redimensionada. Neste caso, **dobra-se** o tamanho dela e se calcula as posições dos elementos já inseridos na nova tabela.

A implementação da **Tabela Hash** desse seguir o seguinte protótipo de funções:

```
typedef struct hash Hash;  
Hash* criaHash(int TABLE_SIZE, int TAMANHO_TIPO);  
void liberaHash(Hash* ha);  
int insereHash(Hash* ha, int chave, void *dados);  
int buscaHash(Hash* ha, int chave, void *dados);
```

Nessa implementação, a **chave** usada na inserção e busca está separada do conjunto de **dados**, e os dados são passados por meio de um **ponteiro genérico**. Isso permite que a Tabela Hash armazene qualquer tipo de dado.

Em seguida, escreva um programa para ler um arquivo contendo os dados de 2287 alunos e os armazene nessa Tabela Hash. Mostre que as operações de inserção e busca funcionam corretamente. Cada linha do arquivo representa um aluno e é composta por 5 informações, separadas por espaço (use a função **fscanf**): matrícula (inteiro), nome e 3 notas (inteiro).



Instruções:

O trabalho será avaliado principalmente levando em consideração:

- 1) Realização das tarefas do trabalho.
- 2) Representação correta da entrada e saída dos dados.
- 3) Uso correto das variáveis e estruturas de dados.
- 4) Uso adequado dos conceitos aprendidos em sala (modularização, hash, árvores, etc, quando for o caso).
- 5) Boa indentação e uso de comentários no código. Evite utilizar comentários excessivamente.

Observações:

- O professor em hipótese alguma irá verificar ou ajudará na construção do código.
- O professor poderá tirar dúvidas sobre o enunciado do problema em horário de aula ou por email.
- A interpretação do problema e a construção da solução fazem parte da avaliação e deverão ser resolvidos pelo aluno.