



Tabela Hash

Dada a implementação da **Tabela Hash** vista em aula, faça adaptações no código de modo a implementar uma **Tabela Hash** com **Encadeamento Separado** para armazenar dados de alunos. Para tanto, use a implementação da **Lista Sequencial Estática** também fornecida.

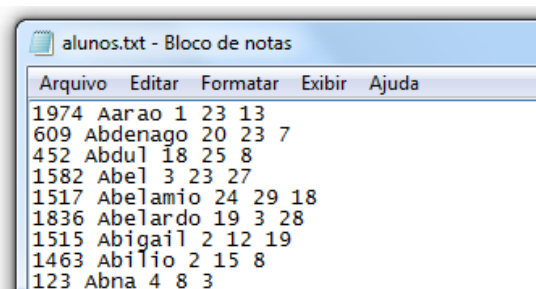
A inserção e busca de um aluno na **Tabela Hash** deverá ser feita por sua matrícula.

- **Inserção:** no caso da posição estar vaga, uma **Lista Sequencial** deverá ser criada e armazenada na **Tabela Hash**. Do contrário, os dados deverão ser armazenados no final da **lista** (colisão).
- **Busca:** no caso da posição estar vaga, o elemento não existe. Do contrário, deverá ser feita uma **busca linear** na **lista**.

Perceba que teremos **duas TAD's** definidas em bibliotecas separadas. Desse modo, a **Tabela Hash** poderá apenas usar as funções da **Lista Sequencial**, isto é, nunca terá acesso direto a estrutura da lista. Quanto ao tamanho da lista, a mesma deve seguir o princípio da realocação:

- Quando criada, seu tamanho será de 10 elementos.
- Sempre que a lista ficar cheia, seu tamanho deverá aumentar em 50%.

Em seguida, escreva um programa para ler um arquivo contendo os dados de 2287 alunos. 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).



Avalie o desempenho da Tabela Hash na inserção desses dados quando diferentes configurações são usadas: tamanho da tabela, tamanho ser primo ou não, função de espalhamento etc. Alguns parâmetros a serem avaliados:

- Número de listas criadas
- Tamanho médio das listas
- Número de colisões
- Desperdício de memória nas listas (posições ociosas)
- Etc