



INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ALGORITMOS E ESTRUTURAS DE DADOS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: TABELA DE ESPALHAMENTO E LISTA ENCADEADA

Prática 06

OBS.: Veja o código que acompanha a prática.

Parte 1: Tabela de Espalhamento Simples

Problema 1: Implemente as funções de espalhamento pedidas em **main.cpp**:

- `hash_function(int)`: recebe inteiro e retorna `long` positivo. Lembre que inteiros podem ser negativos e a função de espalhamento deve retornar um valor > 0 .
- `hash_function(string)`: recebe uma string e retorna `long` positivo. Realize um cálculo processando os caracteres um a um.
- `hash_function(CPF)`: recebe um CPF inteiro e retorna `long` positivo. O CPF internamente é representado como string, mas seu cálculo deve tirar proveito do fato que os caracteres representam dígitos numéricos. Considere que o CPF está sempre correto e não precisa ser verificado.

Rode a prática para ver o resultado dos testes iniciais. Boa funções de espalhamento minimizam colisões. Faça ajustes nas funções ao longo da prática se preciso.

Problema 2: Implemente a tabela de espalhamento simples em **simple_hashtable.h**.

Essa tabela é implementada como um vetor de pares do tipo (chave, valor). O *hash* é calculado sobre a chave e usado para determinar a posição na tabela. Essa versão da tabela não tem a capacidade de lidar com colisões. Caso uma chave com o mesmo *hash* seja inserida, o par (chave, valor) que havia antes deve ser sobrescrito.

Rode e veja os testes da tabela de alunos; o resultado esperado é descrito no código.

Parte 2: Tabela de Espalhamento com Colisões

Problema 1: Implemente a lista encadeada simples em **linkedlist.h**.

Nesta lista simplificada, elementos são adicionados (`add()`) apenas no começo e removidos de qualquer posição (indexada em 0).

Rode e veja o resultado dos testes referentes a lista encadeada.

Problema 2: Implemente a tabela de espalhamento completa em **full_hashtable.h**

Essa tabela de espalhamento lida com colisões usando um vetor de listas encadeadas. As listas contêm pares do tipo (chave, valor). A inserção na tabela acha o índice (usando o *hash*), pega a lista correspondente e insere nela (ou substitui o valor se já existir). Use as funções `search()`, `remove()` e `add()` da lista. A remoção é semelhante.

Rode e veja os resultados das tabelas de notas e de clientes.

Desafio (Opcional): O operador `[]` das tabelas como está permite buscar elementos com facilidade, mas retorna uma cópia *read-only* do valor existente. Altere esses operadores de forma que seja possível usá-los para alterar o valor associado a uma chave na tabela com uma simples atribuição.