

# Hashing com Encadeamento

Sua tarefa neste exercício é implementar *hashing* com a técnica de resolução de colisões chamada *encadeamento*.

Para tanto, considere a seguinte representação de tabela *hash*:

```
typedef struct {
    celula *tb; // tabela hash
    int M;      // tamanho da tabela hash
    int N;      // total de chaves presentes na tabela
} TH;
```

em que `table` é um vetor de nós cabeça da forma

```
typedef struct celula {
    int dado;
    struct celula *prox;
} celula;
```

que representa a tabela *hash*, `M` é o tamanho da tabela e `N` é o total de chaves presentes na tabela.

Você deve submeter um arquivo contendo três funções:

1. A função

```
void THinsere (TH *h, int ch);
```

deve inserir a chave `ch` na tabela *hash*. **A inserção deve ser feita no início da lista encadeada.** *Cuidado para não ter chaves duplicadas.*

2. A função

```
int THremove (TH *h, int ch);
```

deve remover a chave `ch` da tabela *hash*. Esta função deve retornar zero, se a chave foi encontrada e removida, ou `-1` caso a chave não estivesse presente na tabela.

3. A função

```
int THbusca (TH *h, int ch);
```

deve buscar ocorrência da chave `ch` na tabela *hash*. Ela deve retornar 1 se a chave for encontrada, ou 0, caso contrário.

## Restrições

- Considere que o **universo de chaves é composto por inteiros não negativos**.
- Você deve utilizar a *função hash modular* na sua implementação.
- Você deve submeter um código que contenha **apenas as três funções especificadas mais alguma função auxiliar que você julgue necessária à sua solução**. **Você não deve submeter a função `main`.**

*Author: John L. Gardenghi*