

## Estrutura de Dados

### Exercícios: *Listas com encadeamento simples*

## 1 Instruções

1. Desenvolva os exercícios abaixo em C, utilizando como referência as informações apresentadas na aula e a apostila de C que se encontra no Moodle.
2. O objetivo desta lista de exercício é desenvolver um novo tipo *lista* utilizando *structs*, ponteiros e alocação dinâmica de memória.
3. A sua lista deve armazenar valores do tipo inteiro.
4. Crie um módulo `list.h` e `list.c` com guarda de inclusão.
5. Crie um programa cliente para testar a sua nova biblioteca de *listas*.

## 2 Exercícios

1. Utilizando `struct` crie um novo tipo para representar uma célula de uma lista com encadeamento simples.
2. Utilizando `typedef` defina um novo tipo `List` representando uma lista. Se preferir, crie também um *alias* para as células da lista.
3. Faça uma função que cria (aloca dinamicamente) uma nova lista vazia.
4. Implemente uma função com o cabeçalho:

```
Cell* create_cell(int info);
```

que recebe como parâmetro um número inteiro e retorna uma nova célula alocada dinamicamente com o parâmetro `info` armazenado.

5. Utilizando a função do item anterior, faça uma função que insere um novo número no início da lista. A sua função deve ter o seguinte cabeçalho:

```
List* insert_first(List *lst, int info);
```

6. Implemente uma função com o cabeçalho:

```
void print_cell(Cell *c);
```

que recebe como parâmetro uma célula da lista e imprime o número armazenado nela.

7. Utilizando a função do item anterior, faça uma função que imprime a sua lista. A sua função deve ter o seguinte cabeçalho:

```
void print_list(List *lst);
```

8. Crie uma função que desaloca (libera a memória) de uma célula da lista.
9. Utilizando a função do item anterior, faça uma função que desaloca uma lista encadeada por completo.