

## Exercícios 05 :: Lista Sequencial

### Instruções Gerais

- Cada exercício deverá ser escrito em uma ou mais funções, utilizando o editor de sua escolha.
- Implemente os exercícios em C (gcc) ou C++ (g++).
- Ao final, envie o arquivo pelo Moodle.

### Operações sobre Lista Sequencial

Considere o código desenvolvido em aula e disponível no Moodle, que implementa um TAD Lista Sequencial (**struct VectorList**) e algumas operações. A partir do código fornecido, escreva as funções solicitadas nos exercícios para realizar novas operações sobre Lista Sequencial.

1. Escreva uma função que informa (true/1 ou false/0) se a lista está vazia.  
**Função:** `int list_is_empty(VectorList* list);`
2. Escreva uma função que informa (true/1 ou false/0) se a lista está cheia.  
**Função:** `int list_is_full(VectorList* list);`
3. Escreva uma função que retorna a quantidade de posições livres restantes.  
**Função:** `int list_get_available(VectorList* list);`
4. Escreva uma função que remove todos os elementos da lista.  
**Função:** `void list_clear(VectorList* list);`
5. Escreva uma função para remover os últimos N elementos da lista ou o mais próximo disso.  
**Função:** `void list_remove_last(VectorList* list, int n);`
6. Escreva uma função para imprimir a lista de forma reversa.  
**Função:** `void list_print_reverse(VectorList* list);`
7. Escreva uma função que retorna a posição da primeira ocorrência do elemento. Caso o elemento não esteja na lista, a função deve devolver -1.  
**Função:** `int list_find(VectorList* list, int elem);`
8. Escreva uma função para adicionar elementos em lote: recebe um vetor de inteiros, assim como seu tamanho, e os adiciona ao final da lista, enquanto houver espaço.  
**Função:** `void list_add(VectorList* list, int n, int* vet);`

9. Escreva uma função para verificar se a lista está ordenada. Ela deve retornar **true/1** em caso positivo ou **false/0** para o contrário.

**Função:** `int list_is_sorted(VectorList* list);`

10. Escreva uma função para inverter a ordem dos elementos em uma lista. Note que você não deve imprimir a lista, mas sim, inverter seus elementos (no próprio vetor data). Após a chamada à função, a lista deverá estar permanentemente invertida.

**Função:** `void list_reverse(VectorList* list);`

11. Escreva uma função para verificar a igualdade de duas listas. Ela deve retornar **true/1** em caso positivo ou **false/0** para o contrário. OBS: Listas iguais possuem o mesmo tamanho e contêm os mesmos elementos, na mesma ordem.

**Função:** `int list_equal(VectorList* list1, VectorList* list2);`

12. Escreva uma função que cria e devolve uma Lista Sequencial a partir do conteúdo de um array, informado como parâmetro. A Lista Sequencial deve ser criada com um tamanho 2 vezes maior que o do vetor.

**Função:** `VectorList* list_from_vector(int n, int* vet);`

13. Escreva uma função que cria e devolve uma cópia da lista passada. A cópia deve conter os mesmos valores da original.

**Função:** `VectorList* list_copy(VectorList* list);`

14. Escreva uma função que concatena o conteúdo de list2 ao final de list1, obedecendo a capacidade de list1. A função deve retornar a quantidade de elementos adicionados à list1.

**Função:** `int list_concat(VectorList* list1, ListSet* list2);`