**LISTA-10: (Lista)**

1. Crie uma lista de *n* números inteiros, onde *n* e os números devem ser fornecidos pelo usuário.
2. Escreva uma função que verifica se duas listas dadas são iguais (mesmo conteúdo)
3. Considere uma lista contendo números inteiros positivos. Faça uma função que retorne o tamanho da lista.
4. Faça uma função para concatenar duas listas em uma terceira lista que deverá ser posteriormente apresentada ao usuário.
5. Faça uma função que copia uma lista L1 em outra L2, eliminando os elementos repetidos.
6. Dado uma lista que armazene números, escreva uma função que forneça o maior, o menor e a média aritmética dos elementos da lista.
7. Faça uma função para remover os *n* primeiros elementos de uma lista. A função deve retornar se a operação foi possível ou não.
8. Faça uma função para trocar de posição dois elementos de uma lista. A função deve retornar se a operação foi possível ou não.
9. Considere que a struct abaixo está armazenada na sua lista. Faça uma função para buscar o produto de menor preço. A função deve retornar se a operação foi possível ou não.

*struct produto*

*{*

*int codProd; //código do produto*

*std::string nomeProd; //nome do produto*

*float valor; //valor do produto*

*int qtdeEstoque; //quantidade disponível em estoque*

*}*

1. Faça uma função para verificar se uma lista está ordenada ou não (a ordem pode ser crescente ou decrescente).

**DESAFIOS:**

1. Considerando listas de valores inteiros, implemente uma função que receba como parâmetro uma lista e um valor *n* e dívida a lista em duas, de tal forma que a segunda lista começa no primeiro nó após o valor *n*. A função deve retornar à segunda lista.