**LISTA DE EXERCÍCIOS**

1. **Lista simplesmente encadeada:** Faça um método capaz de devolver a quantidade de números maiores que 10 encontradas na lista encadeada. Protótipo: int ListaSimples:: maioresque10();

2. **Lista duplamente encadeada:** siga os passos abaixo para implementar parcialmente a lista que se pede.

1. Apresente a **declaração** de uma classe (somente a estrutura com atributos e protótipos de métodos) para o nó (IntNode), que deve armazenar um inteiro e os ponteiros necessários à lista.
2. Apresente a **declaração** de uma classe para a lista (Int2LList), com todos os atributos e métodos necessários à sua manipulação.
3. **Implemente** um método (mostre o código) para esta classe que realize a busca de um elemento começando, ao mesmo tempo, do final e do início da lista. Se o elemento for encontrado, retornar success, senão, retornar unsuccess.

3. **Fila encadeada:** siga os passos abaixo para implementar parcialmente a fila que se pede.

1. Apresente a **declaração** de uma struct (somente a estrutura com atributos e protótipos de métodos) para o nó (Node), que deve armazenar um inteiro e os ponteiros necessários à fila.
2. Apresente a **declaração** de uma classe para a fila (Fila), com todos os atributos e métodos necessários à sua manipulação.
3. **Implemente** um método (mostre o código) para esta classe que realize a contagem de quantos elementos se encontram na fila.

4. **Deque:** Implemente um deque utilizando um vetor circular.

5. **Lista simplesmente encadeada:** Faça um método capaz de encontrar o elemento do meio da lista (ou a média deles)