

## Exercício 3 - Lista

### Sobre a entrega:

- exercício **INDIVIDUAL**
- submissão no **run.codes** – veja instruções no arquivo **run\_codes.pdf**
- até **25/09/2020**, 23:59h

**Objetivos:** revisar conceitos básicos de lista.

### Parte 1: Implementação

**Problema:** implementar, em linguagem C, o acervo de filmes no sistema Netflix, usando o TAD Lista Não Ordenada visto em aula.

### Simplificações:

- Considere que a lista armazena elementos do tipo **ITEM**, definido da seguinte maneira:
  - *int* chave;           /\*identificador numérico do filme \*/
  - *char* nome[20]; /\*nome do filme\*/

**Implemente** o TAD Lista Não Ordenada, visto nas aulas sobre Listas, utilizando a abordagem de Lista Encadeada Dinâmica. O TAD deve conter, no mínimo, as seguintes funções:

- 1) **lista\_criar**
- 2) **lista\_apagar**
- 3) **lista\_inserir\_pos**
- 4) **lista\_remover**
- 5) **lista\_buscar**
- 6) **lista\_imprimir**
- 7) **lista\_tamanho**
- 8) **lista\_cheia**
- 9) **lista\_vazia**

Utilizando o TAD implementado, desenvolva um programa para armazenar e manipular a lista de filmes do Netflix, considerando a estrutura definida em ITEM.

### Requisitos (obrigatórios):

- Utilize o **.h** do módulo do TAD disponibilizado no Tidia (muito similar ao TAD visto em aula)
- O *Makefile* deve compilar os módulos separadamente, unindo (“link”) apenas na aplicação (conforme apresentado na Aula 3).

### Parte 2: Relatório

Elabore um relatório do exercício em **PDF**, explicando o seu código. Apresente os casos teste que você criou e o objetivo de cada um deles.

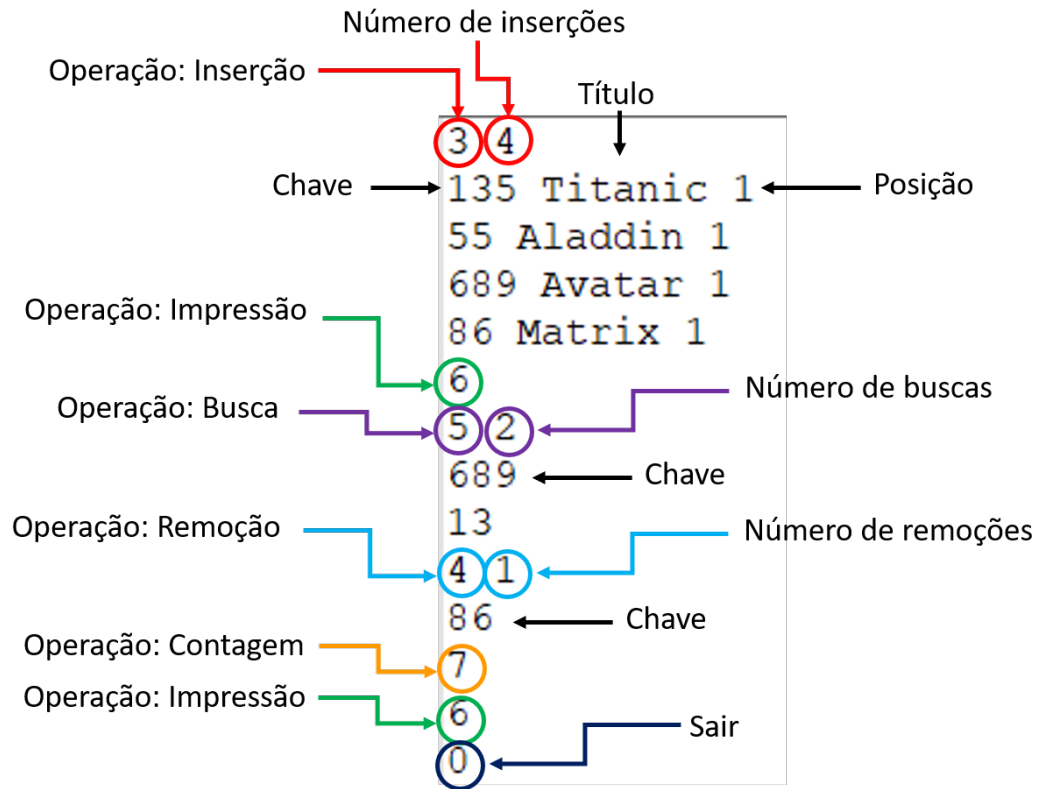
Entregar o arquivo no formato .zip contendo:

- Makefile
- main.c

- lista.c
- lista.h
- Relatório (.pdf)

## Exemplo de Entrada e Saída Esperada

Entrada



Saída

```

1 ← Inseriu
1
1
1
1
86 Matrix
689 Avatar
55 Aladdin
135 Titanic
2 ← Posição da chave buscada
-32000 ← Chave inexistente
1 ← Removeu
3 ← Número de elementos
689 Avatar
55 Aladdin
135 Titanic
  
```

The output shows the results of the operations performed on the list. The first four lines represent the insertion of elements. The next four lines represent the search for the key 135, which is found at position 2. The following line shows the removal of the element with key 86. The next line shows the count of elements, which is 3. The final three lines represent the printing of the list, which now contains the elements with keys 689, 55, and 135.

**IMPORTANTE:**

- ✓ Lembre-se e aplique as “boas práticas de programação”, tais como: organizar e documentar bem o código, definir adequadamente os nomes de variáveis e funções, modularizar o código, identificar autor do código em todos os arquivos, etc...
- ✓ Nomeie os arquivos de modo a facilitar correção do código.
- ✓ No relatório, inclua nome, número USP, identificação do exercício, e apresente as informações requisitadas de modo claro, objetivo e completo.