

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

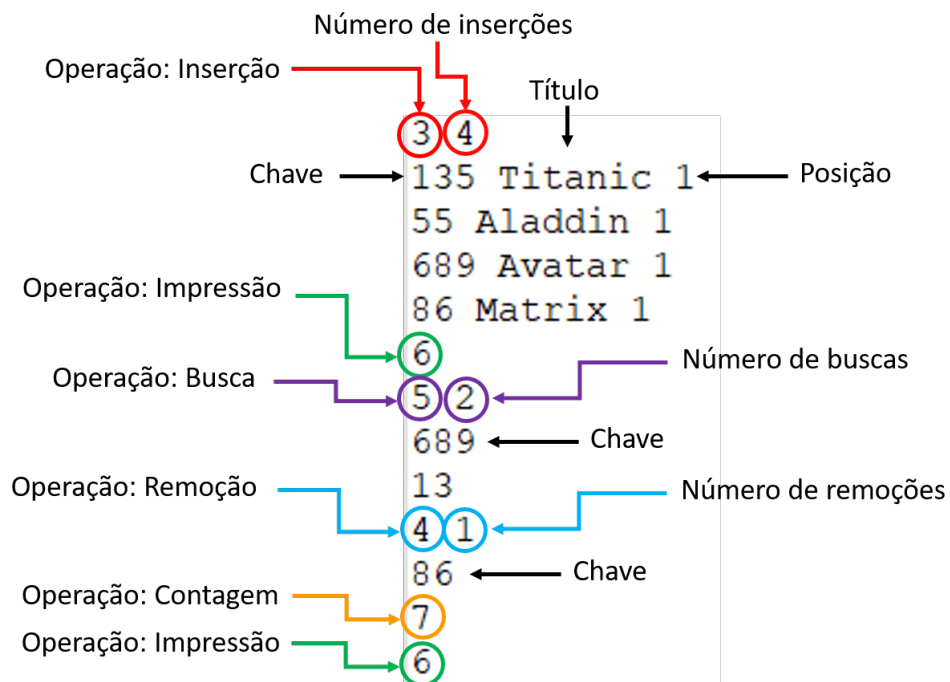
Elabore um relatório do exercício em **PDF**, explicando o seu código. Apresente os casos teste que você criou e o objetivo de cada um deles.

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

```

Impressão

Impressão

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.