



INSTITUTO FEDERAL
Maranhão
Campus Caxias

Curso: Ciência da Computação
Professor: Luís Fernando Maia

Nota

Nome: _____

Disciplina: _____ Data: ____/____/____

Exercícios 04 - Listas Encadeadas

1. Faça um programa que leia um conjunto de números inteiros e armazene-os ordenadamente em uma lista encadeada simples.
2. Crie uma lista encadeada dupla para armazenar os dados de alunos em um determinado curso. A estrutura deve conter a matrícula do aluno, nome e as notas da primeira, segunda e terceira provas. O programa deve realizar as operações abaixo:
 - a. Cadastrar as informações dos alunos.
 - b. Buscar um aluno por sua matrícula e imprimir seus dados.
 - c. Informar o nome do aluno com maior média geral.
 - d. Para cada aluno, reportar se ele foi aprovado ou reprovado, considerando a média 7 como aprovativa.
 - e. Criar uma nova lista com os alunos organizados por ordem alfabética.
3. Imagine que você está desenvolvendo um player de música que utiliza listas circulares para representar as playlists de acordo com as estruturas de dados abaixo. Sabendo que seu player pode gerenciar duas playlists ao mesmo tempo, implemente os procedimentos abaixo.
 - a. Adicionar uma nova música ao fim de uma playlist. Observe que é necessário ler nome e duração de cada arquivo, atualizando a duração e quantidade na playlist.
 - b. Buscar pelo nome e remover a música da playlist. Observe que é necessário atualizar a duração e quantidade de músicas na playlist.
 - c. Combinar duas playlists de música em uma só. Neste caso uma playlist é adicionada ao fim da outra, ficando uma delas com todas as músicas e a outra vazia.
 - d. Imprimir os dados das playlists, que são a quantidade de músicas que cada uma possui, os nomes das músicas e sua duração total.

```
typedef struct{  
    int h;  
    int m;  
    int s;  
} tempo;
```

```
typedef struct cel {  
    char arquivo[50];  
    tempo duracao;  
    cel *ant;  
    cel *prox;  
} musica;
```

```
typedef struct {  
    musica *primeira;  
    int quantidade;  
    tempo duracao;  
} playlist;
```