

### **LISTA DE EXERCÍCIOS**

- Alocação Dinâmica em Memória -

#### **LIMITE PARA ENVIO: Data da Prova #03**

**Enviar as soluções em arquivo compactado para o e-mail: [adriano.exe@gmail.com](mailto:adriano.exe@gmail.com)**

#### **Questão 01 [Listas Encadeadas]**

Considere um site de busca de conteúdos que possua uma coleção de arquivos *torrent* disponibilizados para *download* dos usuários. Considerando que as informações dos arquivos *torrent* (nome do conteúdo e link para download) são armazenados em uma estrutura de dados do tipo lista simplesmente encadeada, faça um programa que dado o nome de um conteúdo, busque e imprima o *link* correspondente, e após isso, mova o nó que contém o nome buscado para o início da lista, de forma que este possa ser encontrado mais rapidamente na próxima vez que for buscado.

Faça também uma função para cadastro de arquivos no site e uma função para imprimir todos os arquivos da mantidos na lista.

#### **Questão 02 [Exclusão em uma Lista]**

Faça um programa que armazene dinamicamente (através de estrutura de lista encadeada) registros de Artigos Científicos de um Departamento (Autor, Título, Área de Conhecimento, Ano de Publicação).

- Desenvolva uma função que retorne a quantidade de registros alocados na lista;
- Desenvolva uma função que realize a impressão do registro que esteja o mais próximo possível do meio da lista.
- Desenvolva uma função que realize a impressão de todos os registros cadastrados.
- Desenvolva uma função que realize a impressão somente das publicações entre um período de anos escolhidos pelo usuário.

*Obs: Os requisitos C e D devem compartilhar a mesma rotina de impressão de dados.*

- Desenvolva uma função que realize a **exclusão** de um determinado registro da lista. Para isto, solicite ao usuário o “**Título do Artigo**” a ser eliminado.

*Obs.: Não se esqueça de utilizar o comando free(ponteiro) para realizar a liberação de memória.*

#### **Questão 03 [Fila]**

Escreva um programa que simule o controle de uma pista de decolagem de aviões em um aeroporto. Neste programa, o usuário deve ser capaz de realizar as seguintes tarefas:

- Adicionar um avião na fila para decolagem;
- Listar os aviões aguardando na fila de decolagem;
- Autorizar decolagem de avião (exclusão do primeiro da fila);
- Abortar a decolagem de um avião (alterá-lo da posição inicial para a posição final da fila);
- Exibir as características do primeiro avião da fila;

Considere que os aviões possuem as seguintes informações: NUMERO do voo, cidade de ORIGEM e cidade de DESTINO, quantidade de PASSAGEIROS.

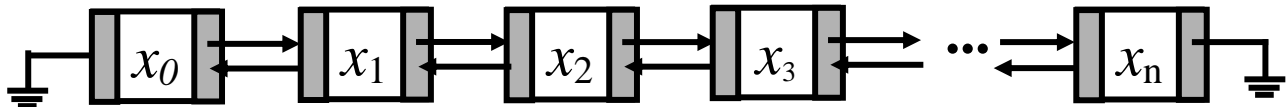
#### Questão 04 [Lista Ordenada]

Crie um programa que cadastre dinamicamente (através de lista encadeada) e **ordenadamente**, os produtos de uma loja de departamentos (código de barras, descrição do produto, quantidade em estoque, valor de venda).

- Faça uma função para imprimir a relação de todos os produtos.
- Faça uma função para exclusão de um determinado item, através do código de barras.

#### Questão 05 [Lista Duplamente Encadeada]

Faça a implementação de um programa utilizando uma estrutura do tipo “Lista Duplamente Encadeada”. Nesta abordagem, cada nó/registro contém o endereço para o nó/registro seguinte e também para o nó/registro anterior.



- Faça uma função de cadastro, para inclusão no início da lista;
- Faça uma função de cadastro, para inclusão ao final da lista;
- Faça uma função para exclusão do primeiro elemento da lista;
- Faça uma função para exclusão do último elemento da lista;
- Faça uma função para impressão de toda a lista, a partir do primeiro elemento;
- Faça uma função para impressão de toda a lista, a partir do último elemento;

#### Questão 06 [Listas e Sub-Listas]

Crie um programa que armazene dinamicamente informações de veículos e suas multas.

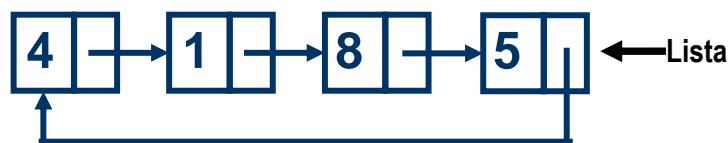
Veículo = Modelo, Placa, Ano, (e Apontador para Multas).

Multas = Data, Código da Infração, Valor Multa, (e Apontador para Próxima Multa);

- Crie uma função para realizar o cadastro de veículos;
- Crie uma função para realizar o cadastro de multas de um determinado carro através da placa;
- Crie uma função que leia do usuário uma placa, e imprima os dados referentes a esse carro, suas multas e o valor total de multas.

#### Questão 07 [Lista Circular]

Faça a implementação de um programa utilizando uma estrutura de Lista Encadeada Circular. Nesta metodologia, o último nó da lista sempre deverá apontar para o primeiro nó (registro) da lista. Portanto, para realizar operações de Inclusão e Consulta, basta oferecer como parâmetro o ponteiro do Final da Lista.



- Faça uma função para cadastro na Lista Circular.
- Faça uma função para impressão da Listagem Geral;
- A lista circular trata a estrutura de lista encadeada como pilha, fila ou ambas? Por quê?

### Questão 08 [Lista Heterogênea]

Uma lista heterogênea é aquela capaz de armazenar nós com tipos distintos entre si.

Diversas aplicações precisam construir listas heterogêneas, pois necessitam agrupar elementos afins, mas não necessariamente iguais.

Considere uma aplicação que necessita calcular a área total de diferentes formas geométricas, como retângulos (base e altura), triângulos (lado a, lado b e lado c) e círculos (raio), agrupados em uma única lista encadeada.

Observe que os dados necessários para cada forma geométrica, e o cálculo para se obter a área são distintos entre si.

Para criar uma lista heterogênea, devemos definir 03 novos tipos de *struct*, representando cada forma geométrica, e outro tipo *struct* para representar, de forma genérica, uma forma geométrica, como por exemplo:

```
typedef struct FormaGeometrica{
    int tipo;           //se 1: retângulo, se 2: triângulo, se 3: círculo;
    void* info;         //ponteiro genérico que aponta para qualquer uma das três structs
    struct FormaGeometrica prox; //ponteiro para próxima forma
}FormaGeometrica;
```

Faça um programa que, utilizando uma única **Lista Heterogênea**, apresente as seguintes opções ao usuário:

- 1 - Cadastrar Retângulo
- 2 - Cadastrar Triângulo
- 3 - Cadastrar Círculo
- 4 - Obter Área dos Retângulos
- 5 - Obter Área dos Triângulos
- 6 - Obter Área dos Círculos
- 7 - Obter Área Total Cadastrada

*Obs.: Utilize a Fórmula de Heron para calcular a área do triângulo.*

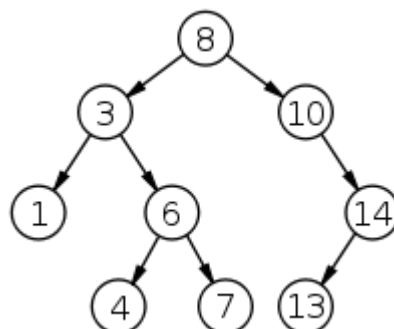
### Questão 09 [Árvore Binária]

Uma **árvore binária** (ou árvore de busca binária, ABB) é uma estrutura de dados onde todos os nós (*structs*) possuem um apontador para o vizinho à esquerda (sub-árvore esquerda) e um apontador para um vizinho à direita (sub-árvore direita).

A organização da árvore binária é feita de modo a **otimizar a busca de informações na estrutura**, sendo assim estruturada... O valor chave da sub-árvore esquerda possui valor inferior ao do nó raiz e o valor chave da sub-árvore direita possui um valor superior ao nó raiz (esta é a forma padrão, podendo as sub-árvores serem invertidas, dependendo da aplicação).

O objetivo desta árvore portanto, é estruturar os dados de forma flexível, permitindo [busca binária](#).

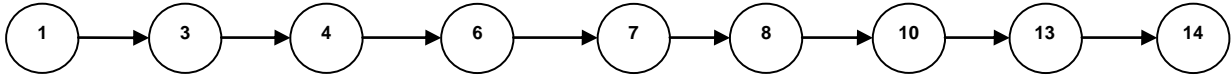
Observe o exemplo da árvore abaixo...



Observe...

- Todos os nós a esquerda do nó raiz (valor 8) possuem valor inferior a ele...
- Todos os nós a direita do nó raiz (valor 8) possuem valor superior a ele...
- Todos os nós a esquerda do nó com valor 3, possuem valor inferior a ele...
- Todos os nós a direita do nó com valor 3, possuem valor superior a ele...

Se essa estrutura fosse construída como uma lista ordenada convencional, teríamos algo como...



Entretanto, na estrutura de uma lista ordenada convencional, para acessar ao registro “13” teríamos que percorrer 7 nós antes de chegar ao alvo... Enquanto que na estrutura de Árvore Binária, teríamos que percorrer apenas 3 nós (8, 10, 14).

Faça a implementação de uma árvore de busca binária:

- a) Função para cadastrar registros.
- b) Função para localizar registros (imprimir quantos saltos foram necessários para localizar).