

Lista de AED

- 1) Hoje em dia, os clubes de futebol monitoram seus atletas durante o jogo, para monitorar seu desgaste durante a partida. Faça uma lista em que seja guardado o nome e distância que cada jogador percorreu por jogo. Os dados deverão ser salvos em um arquivo "time.txt". No final liste o jogador que percorreu maior distância e o jogo onde ele mais correu.
- 2) Faça um programa que guarde uma lista com os modelos de carros, consumo de cada um, e a quilometragem que andou. Calcule e mostre no menu:
 - a. O modelo do carro mais econômico, ou seja, o que tem menor consumo;
 - b. Quantos litros de combustível o carro selecionado consome para percorrer uma distância de X quilômetros e quanto isto custará, considerando um que a gasolina custe R\$ 4,25 o litro.
- 3) Foram anotadas as idades e alturas de alunos do curso de Engenharia de Redes. Faça um programa que guarde em uma lista o nome e altura de cada aluno, outra lista com todos os alunos abaixo da média do curso, e uma com os alunos acima da média.
- 4) Crie um programa orientado a objetos que testa se duas listas lineares são iguais. Lembre-se que duas listas são iguais se possuem os mesmos elementos na mesma ordem e não somente os mesmos elementos.
- 5) Implemente um programa em C++ orientado a objetos que realiza uma busca binária em um vetor ordenado com 10 números, e mostre na tela o passo a passo dessa busca. É preferível que use `system("CLS")`; da biblioteca `<cstdlib>` para limpar a tela em cada passo da busca.
- 6) Suponha que o vetor `v` tem 511 elementos e que `x` não está no vetor. Quantas vezes, exatamente, a função busca Binária vai comparar `x` com um elemento do vetor? Suponha agora que o vetor `v` tem 50000 elementos e que `x` não está no vetor. Quantas vezes, aproximadamente, a função busca Binária vai comparar `x` com um elemento do vetor?
- 7) O professor corrigiu as provas e vai lançar as notas na ordem em que os alunos sentam na sala, para fim de analisar se determinada nota que deseja buscar se localiza de um lado ou de outro da sala. Para isso ele vai lançar as notas de 0 a 10 `N` vezes, ($N < 20$) e após isso utilizar da busca binária, em que a mediana da posição das notas na lista separa o lado esquerdo da sala do lado direito, para informar se a nota `X` que ele procura é de algum aluno que senta do lado direito ou do lado esquerdo.

8) Um Hospital precisa organizar sua fila de atendimento, organizando por nome e risco do paciente. Faça um programa orientado a objetos que cadastre os pacientes, e organize o atendimento por ordem de risco em Vermelho(Crítico), Amarelo (Urgente), Verde(Sem prioridade).

9) Uma empresa de venda de carros, motos e caminhões necessita de um programa em C++ orientado a objetos e que utilize arquivos como banco de dados com explicações de cada tipo de veículo. O programa deve ter uma classe principal **Class veiculos**{ com os seguintes atributos: motor, transmissão, cor, num de lugares }, e uma classe para cada tipo de veículo herdando os atributos dessa classe principal e com seus atributos particulares de cada tipo de veículo.

O menu deve conter:

1- Cadastrar um novo carro, moto ou caminhão.

2- Ver catálogo de carros, motos ou caminhões.

3- Comprar carro, moto ou caminhão.

4- sair.

10) Você é um engenheiro que foi contratado para fazer um programa que ajuda um mercado a cadastrar todos os produtos que ele trabalha (no mínimo 10 e no máximo 15). Seu programa deve ter um menu que seja possível fazer o cadastramento manual dos produtos, sempre verificando se já chegou ao limite máximo e deve ter uma encontra-se cadastrado.

11) Você está trabalhando em uma loja de materiais. Toda semana irá receber novas encomendas de tijolos de vários tamanhos em grandes caixas e precisa saber onde deve armazenar esses produtos. Você recebe todos em caixas etiquetadas com o lote dos produtos. Por exemplo, duas caixas com os respectivos intervalos (3, 5) e (7, 9) representam lotes de produtos {3, 4, 5, 7, 8, 9}. Um número pode estar presente em mais de uma caixa, e ele aparece na lista uma vez para cada intervalo. Por exemplo, três caixas com os respectivos intervalos (3, 5), (4, 8) e (5,9) representam lotes {3, 4, 4, 5, 5, 5, 6, 6, 7, 7, 8, 8, 9}. A entrada contém vários casos de teste e termina quando não for achado o lote que se busca. Cada caso de teste é composto por várias linhas. A primeira linha contém um inteiro positivo X ($X < 100$) que indica a quantidade de caixas. Seguem X linhas, cada uma delas com dois valores W e Z ($1 \leq W < Z \leq 100$) que representam os lotes presentes naquela caixa. A próxima linha de entrada conterá um único inteiro positivo J ($1 \leq J \leq 100$), que indica o número que deseja pesquisar linearmente após todos os produtos serem organizados na estante ou rack. Obs. A

estante ou rack possui prateleiras enumeradas de 0 a K, sendo que este K não deve ser superior a 1000. A saída deve indicar se o J foi encontrado e em qual posição foi encontrado primeiramente.

12) Um banco precisa desenvolver um servidor cadastre contas bancárias. A empresa em que você trabalha foi contratada, e o designou para implementar em C++ um programa orientado a objetos e com estrutura de listas duplamente encadeada. O programa deve conter um menu com os seguintes itens:

- 1- Cadastrar contas bancárias.
- 2- Imprimir conteúdo armazenado no servidor.
- 3- Procurar conta.
- 4- Retirar contas.
- 5- Imprimir o servidor após a modificação.
- 6- sair.

As informações necessárias de uma conta bancária são **Nome do banco, Nome do cliente, Cpf, Agência, Número da conta e senha.**

O programa também deve conter uma restrição de acesso de usuários para chamar os métodos dos casos 1 e 4, com login padrão: **admin** senha: **admin**.

13) Dada uma pilha de n cartas numeradas de 1 até n com a carta 1 no topo e a carta n na base. A seguinte operação é realizada enquanto tiver 2 ou mais cartas na pilha.

Jogue fora a carta do topo e mova a próxima carta (a que ficou no topo) para a base da pilha.

Sua tarefa é encontrar a sequência de cartas descartadas e a última carta remanescente.

Cada linha de entrada (com exceção da última) contém um número $n \leq 50$. A última linha contém 0 e não deve ser processada. Cada número de entrada produz duas linhas de saída. A primeira linha apresenta a sequência de cartas descartadas e a segunda linha apresenta a carta remanescente.

A entrada consiste em um número indeterminado de linhas contendo cada uma um valor de 1 até 50. A última linha contém o valor 0.

Para cada caso de teste, imprima duas linhas. A primeira linha apresenta a sequência de cartas descartadas, cada uma delas separadas por uma vírgula e um espaço. A segunda linha apresenta o número da carta que restou. Nenhuma linha tem espaços extras no início ou no final. Veja exemplo para conferir o formato esperado.

- 14)** Um PetShop precisa organizar seus clientes de forma em que estejam guardadas informações como Nome do Animal, Dono, quantidade de serviços que ele realizou dentro da empresa. Desta forma eles conseguem visualizar quais são os clientes mais fiéis. Faça um programa que guarde estas informações em um arquivo “pet.txt”, e que faça uma busca com sentinela e mostre qual o cliente realizou mais serviços pelo PetShop.
- 15)** Em um restaurante, as mesas são organizadas na forma de uma matriz 10x10, e cada mesa contém os produtos consumidos com valor de cada um, nome do cliente e garçom que está atendendo. Faça um programa em que um garçom possa adicionar produtos a uma mesa, e buscar estes por meio de uma busca com sentinela. Os produtos devem estar previamente cadastrados, ou senão o usuário poderá cadastrar dentro do mesmo.
- 16)** Implemente uma função que ordene uma lista de nomes próprios, trocando os nomes entre os blocos, em seguida imprima na tela a lista antes e depois de ordenada.
- 17)** Você deve fazer duas listas e ordená-las, uma de Nomes masculinos e outra de nomes femininos. Implemente uma função em C++ que faça a união das duas listas, mantendo a ordenação.
- 18)** Desenvolver um programa que implemente uma Lista Ordenada para frutas, com funções para:
- Inserir elementos
 - Mostrar Elementos
 - Remover um elemento
 - Pesquisar elemento