

1 - O Prefeito de São Luís deseja encontrar qual o melhor lugar para construir uma escola com capacidade  $N$  de discentes. O que o prefeito tem em mãos é um mapa que diz onde está cada aluno da cidade. Sua tarefa é construir um programa que indique onde é o melhor lugar de se construir a escola, considerando que o melhor lugar é aquele que preencha as vagas da escola levando em consideração a menor distância em relação aos alunos. A entrada do programa consiste na localização (2D) no mapa das casas dos alunos, assim como a quantidade de alunos dentro da casa e a capacidade total ( $N$ ). No máximo, o mapa possui 1000 entradas. Inicialmente o usuário informará a quantidade de entradas (inferior a 1000) e em seguida a posição ( $X, Y$ ) e a quantidade de alunos ( $Q$ ) de cada casa.

2 - Um professor deseja calcular a nota das provas dos alunos da seguinte maneira: o aluno que recebeu a melhor avaliação terá sua nota mudada para 10. O segundo melhor, terá uma proporção da melhor nota de 10. Até que o último, que obteve a pior avaliação, terá nota 6.0. Escreva um programa que receba os seguintes dados: quantidade de alunos, nome dos alunos, nota de cada um deles. Depois calcule as novas notas conforme o critério solicitado e exiba na tela os resultados.

3 - Escreva um programa que permita o usuário carregar palavras e seus significados. O usuário irá definir quantas palavras deseja carregar. Em seguida, permita ao usuário fazer buscas nesse dicionário, onde ele deve informar parte (ou toda) da palavra que deseja saber o significado.

4- Um usuário possui um estacionamento de carros e deseja organizá-los em uma lista. Cada carro para ele possui uma placa, modelo, ano e proprietário. Sua tarefa consiste em implementar um programa que permita cadastrar todos os carros na lista e depois permita fazer consultas pelos termos de buscar placa, ou modelo ou proprietário.

5- Implemente uma função que una duas listas de inteiros em apenas uma lista sem repetições>

6- Implemente uma função de inserção ordenada de elementos numa lista de inteiros

7- A partir de (6) implemente uma função que realize a inserção ordenada de palavras numa lista de strings

8- Implemente um programa que permita ao usuário informar um número qualquer de palavras e seus significados. Cada um dos pares deverá ser guardado numa lista. Depois o usuário deseja encontrar as palavras através de métodos de busca pela palavra ou pelo significado

9- Dado uma coleção qualquer de pontos cartesianos, escreva um algoritmo que calcule o menor quadrado envolvente destes pontos.

10 - Dado uma coleção qualquer de pontos, escreva um algoritmo que calcule o menor círculo envolvente destes pontos.

11 - Uma pessoa desorganizada guarda todos os comprovantes de pagamentos em um arquivo texto imenso que contém a seguinte estrutura: Descrição\_Conta; Valor; Data\_Pagamento; Número\_do\_comprovante. Imaginando que o dono do arquivo irá digitar na entrada padrão cada linha desse arquivo, faça um programa que informe ao dono desorganizado, o valor gasto em cada mês/ano.

12 - Você foi contratado para gerar um vetor de estruturas Caminho. A estrutura Caminho tem dois parâmetros: distância e um ponteiro para outro Caminho que tenha valor de distância maior que o seu próprio. Você foi contratado para escrever uma função que receba um vetor de distâncias (float) e precisa que você retorne um vetor de estrutura Caminho. Toda memória deve ser alocada na forma de somente 1 vetor de estruturas Caminho.

13 - Fazer um programa para simular uma agenda de telefones em arquivo. Para cada pessoa deve-se ter os seguintes dados: Nome, E-mail, Endereço (contendo campos para Rua, numero, complemento, bairro, cep, cidade, estado, país), Telefone (contendo campo para DDD e número), Data de aniversário (contendo campo para dia, mês, ano), Observações : Uma linha (string) para alguma observações especial. O programa deve ser capaz de cadastrar contato, remover contato, consultar contato ou editar contato.

14 - Escreva uma função para ordenar palavras em ordem alfabética crescente. As palavras possuem tamanho variável. Cada entrada é constituída por um identificador ID e por uma palavra word (ou seja, vai ser uma struct). A saída deve apresentar cada ID de entrada juntamente com a palavra correspondente word ordenados alfabeticamente. Obs.: Os caracteres possíveis são [a-z] e [A-Z]