

Algoritmos e Estruturas de Dados I

Quarta Lista de Exercícios

1. STRINGS

1. Ler nome, sexo e idade. Se sexo for feminino e idade menor que 25, imprime o nome da pessoa e a palavra “ACEITA”, caso contrário imprimir “NÃO ACEITA”.

2. Faça um programa que conte o número de 1's que aparecem em uma string. Exemplo: “0011001” -> 3

3. Escreva um programa que leia a idade e o primeiro nome de 10 pessoas. Seu programa deve terminar quando uma idade negativa for digitada. Ao terminar, seu programa deve escrever o nome e a idade das pessoas mais jovens e mais velhas.

4. Faça um programa que preencha uma matriz de string com os modelos de cinco carros (exemplos de modelos: Fusca, Gol, Vectra, etc.). Em seguida, preencha um vetor com o consumo desses carros, isto é, quantos quilômetros cada um deles faz com um litro de combustível. Calcule e mostre:

- a) O modelo de carro mais econômico;
- b) Quantos litros de combustível cada um dos carros cadastrados consome para percorrer uma distância de 1.000 quilômetros.

5. O código de César é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de três posições, ‘A’ seria substituído por ‘D’, ‘B’ se tornaria ‘E’, e assim por diante. Implemente um programa que faça uso desse código de César (3 posições), entre com uma string e retorne a string codificada. **Exemplo:**

- **String:** a ligeira raposa marrom saltou sobre o cachorro cansado
- **Nova string:** d oljhlud udsrsvd pduurp vdowrx vreuh r fdfkruur fdqvdgr

6. Faça um programa que, dada uma string, diga se ela é um palíndromo ou não. Lembrando que um palíndromo é uma palavra que tenha a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita. **Exemplo:**

- ovo
- arara
- socorram-me, subi no ônibus em Marrocos
- anotaram a data da maratona

7. Construa um programa que leia duas strings fornecidas pelo usuário e verifique se a segunda string lida está contida na primeira string, ou seja, se a segunda string é uma substring da primeira, retornando o resultado da verificação.

8. Faça um programa para ler uma tabela contendo os nomes dos alunos de uma turma de 5 alunos. O programa deve solicitar ao usuário os nomes dos alunos. Uma vez lidos todos os alunos, o usuário irá indicar um nome que ele deseja verificar se está presente na lista, onde o programa deve procurar pelo nome (ou parte deste nome) e se encontrar deve exibir na tela o nome completo e o índice do vetor onde está guardado este nome.

9. Leia duas cadeias de caracteres A e B. Determine quantas vezes a cadeia A ocorre na cadeia B.
10. Faça um programa que contenha um menu com as seguintes opções:
- a) Ler uma string S1 (tamanho máximo 20 caracteres);
 - b) Imprimir o tamanho da string S1;
 - c) Comparar a string S1 com uma nova string S2 fornecida pelo usuário e imprimir o resultado da comparação;
 - d) Concatenar a string S1 com uma nova string S2 e imprimir na tela o resultado da concatenação;
 - e) Imprimir a string S1 de forma reversa;
 - f) Contar quantas vezes um dado caractere aparece na string S1. Esse caractere deve ser informado pelo usuário;
 - g) Substituir a primeira ocorrência do caractere C1 da string S1 pelo caractere C2. Os caracteres C1 e C2 serão lidos pelo usuário;
 - h) Verificar se uma string S2 é substring de S1. A string S2 deve ser informada pelo usuário;
 - i) Retornar uma substring da string S1. Para isso o usuário deve informar a partir de qual posição deve ser criada a substring e qual é o tamanho da substring.

2. STRUCTS – ESTRUTURAS/REGISTROS

1. Implemente um programa que leia o nome, a idade e o endereço de uma pessoa e armazene os dados em uma estrutura.
2. Construa uma estrutura aluno com nome, número de matrícula e curso. Leia do usuário a informação de 5 alunos, armazene em vetor dessa estrutura e imprima os dados na tela.
3. Faça um programa que armazene em um registro de dados (estrutura composta) os dados de um funcionário de uma empresa, compostos de: Nome, Idade, Sexo (M/F), CPF, Data de Nascimento, Código do Setor onde trabalha (0-99), Cargo que ocupa (string de até 30 caracteres) e Salário. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela.
4. Utilizando uma estrutura, faça um programa que permita a entrada de nome, endereço e telefone de 5 pessoas e os imprima em ordem alfabética.
5. Faça um programa que leia os dados de 10 alunos (Nome, matrícula, Média Final), armazenando em um vetor. Uma vez lidos os dados, divida estes dados em 2 novos vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a média mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.
6. Faça um programa que leia um vetor com dados de 5 livros: título (máximo 30 letras), autor (máximo 15 letras) e ano. Procure um livro por título, perguntando ao usuário qual título deseja buscar. Mostre os dados de todos os livros encontrados.
7. Faça um programa que seja uma agenda de compromissos e:
 - a) Crie e leia um vetor de 5 estruturas de dados com: compromisso (máximo 60 letras) e data. A data deve ser outra estrutura de dados contendo dia, mês e ano.
 - b) Leia dois inteiros M e A e mostre todos os compromissos do mês M do ano A. Repita o procedimento até ler M = 0.

8. Fazer um programa para simular uma agenda de telefones. Para cada pessoa deve-se ter os seguintes dados:

- Nome;
 - E-mail;
 - Endereço (contendo campos para Rua, número, 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 observação especial;
 - Crie tantas estruturas quanto forem necessárias para melhor representar o problema;
-
- a) Definir a estrutura acima.
 - b) Declarar a variável agenda (vetor) com capacidade de agendar até 100 nomes.
 - c) Definir um bloco de instruções busca por primeiro nome: Imprime os dados da pessoa com esse nome (se tiver mais de uma pessoa, imprime para todas).
 - d) Definir um bloco de instruções busca por mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse mês.
 - e) Definir um bloco de instruções busca por dia e mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse dia e mês.
 - f) Definir um bloco de instruções insere pessoa: Insere por ordem alfabética de nome.
 - g) Definir um bloco de instruções retira pessoa: Retira todos os dados dessa pessoa e desloca todos os elementos seguintes do vetor para a posição anterior.
 - h) Definir um bloco de instruções imprime agenda com as opções:
 - Imprime nome, telefone e e-mail.
 - Imprime todos os dados.
 - i) O programa deve ter um menu principal oferecendo as opções acima.