



Exercício de Programação Estruturada

- 1) Defina as seguintes estruturas e mostre exemplos de como são utilizadas em C: Do-While, For e Switch-case !
- 2) Defina o que são e para que serve as Funções na linguagem C ?
- 3) Fale sobre a regra de escopo de uma função e defina o que é variável global e local.
- 4) Explique quais as diferenças entre passagem de parâmetros por referência e por valor.
- 5) Defina o que é recursividade e implemente uma função recursiva que calcule o fatorial de um número inteiro.
- 6) O que são Vetores e Strings em C, Exemplifique as regras sintáticas de declaração e manipulação de ambos !
- 7) O que são ponteiros em C ? Mostre um exemplo prático da utilização de ponteiros.
- 8) Crie um programa que armazene 10 números em um vetor A e gere um vetor B cujos elementos consistem no quadrado dos elementos de A.
Exemplo:
A[1] = 4 B[1] = 16
A[2] = 3 B[2] = 9
A[3] = 6 B[3] = 36
- 9) Um programa em C contém a seguinte definição de array:

```
int a[8] = {10, 20, 30, 40, 50, 60, 70, 80}
```

 - (a) O que representa a ?
 - (b) O que representa a + 2?
 - (c) Qual é o valor de *a?
 - (d) Qual é o valor de *a + 2?
 - (e) Qual é o valor de *(a + 2) ?
- 10) Dadas as seguintes iniciações:

```
int ar[] = {10, 15, 4, 25, 3, -4};  
int *p = &ar[2];
```

Quais são os resultados das avaliações das seguintes expressões?

 - (a) *(p + 1)
 - (b) p[-1]
 - (c) ar[*p++]
 - (d) *(ar + ar[2])

- 11) O que será escrito na tela após a execução do trecho de programa a seguir?

```
char ar[] = "Boa noite", *ptr = &ar[1];  
printf("%s", ptr + 3);
```

- 12) Interprete a definição de variável a seguir:

```
char *ar[] = {"azul", "vermelho", "branco"};
```

- 13) Faça um programa que receba o nome e as duas notas de seis alunos e mostre o relatório abaixo. Considere que a média mínima para aprovação é 7,0.

Aluno	Prova 1	Prova 2	Média	Situação
João	8,0	5,0	6,5	Reprovado
Carla	10,0	9,0	9,5	Aprovado

- Média da classe: ?
 - Quantidade de aprovados: ?%
 - Quantidade de reprovados: ?%
- 14) Escreva um programa que leia uma string do usuário e verifique se ela é um palíndromo (ou seja, se pode ser lida da mesma forma da esquerda para a direita e vice-versa).
- 15) (a) Escreva uma função em C com dois parâmetros: (1) um parâmetro do tipo **double** e (2) um parâmetro do tipo **int**. Essa função deverá retornar o valor do primeiro parâmetro elevado ao segundo. Em outras palavras, se o primeiro parâmetro é denominado x e o segundo é denominado n , essa função deverá retornar o resultado de x^n . (b) Escreva um programa em C que receba como entradas um valor real x e um valor inteiro n , utilize a função descrita em (a) para calcular x^n e exiba esse resultado na tela. (c) Escreva uma função recursiva que substitua a função iterativa implementada no item (a).
- 16) Escreva um programa que leia um vetor de números inteiros. (a) Faça uma função que ordene seus elementos em ordem crescente. (b) faça uma função que ordene em ordem decrescente. (c) Faça uma função recursiva que busque um elemento qualquer nesse vetor.
- 17) Faça um programa que leia uma matriz 3x4 de inteiros e calcule a soma dos elementos de cada coluna, armazenando o resultado da soma em um vetor de 4 elementos. Por fim, o programa deverá exibir o conteúdo de tal vetor.
- 18) Crie um programa que leia uma matriz 3x3 e troque os elementos da linha 2 pela coluna 2 e vice-versa.
- 19) (a) Escreva uma função que receba um vetor de inteiros como parâmetro, percorra tal vetor e substitua os valores pares por 1 e os ímpares por 0. As alterações realizadas devem ter impacto no arranjo passado como parâmetro real. (b) Escreva uma função `main()` que

leia um vetor de 10 números inteiros, faça uma cópia de tal vetor e exiba na tela o vetor antes e depois da chamada da função descrita no item (a).

20) Faça um programa que tenha o seguinte menu:

1 – Incluir pessoa; 2 – Listar pessoas; 3 – Remover pessoa

Quando o usuário selecionar a opção 1, o programa pede o nome da pessoa para ser adicionado. Quando o usuário digitar a opção 2, o programa lista todas as pessoas cadastradas. Quando o usuário selecionar a opção 3, o programa pede o nome da pessoa que deseja remover. **Dica:** Para comparar duas *strings*, utilize a função `strcmp`, presente na biblioteca `string.h`. O código abaixo ilustra a comparação de *strings* utilizando essa função.