

2ª Parte da Lista de Exercícios

Prof. Mateus Ferreira Satler

DECSI – ICEA - UFOP

Obs.: para todos os exercícios de codificação (criação de código na linguagem C) a seguir, crie 1 projeto separado para cada exercício (não faça mais de um exercício dentro do mesmo código).

PARTE VII – LAÇOS ENCAIXADOS

21. Construa um programa que imprima todas as possíveis jogadas para quatro dados normais, com faces de 1 a 6, sem que haja, em cada jogada, repetição de valores para dados distintos.

22. Construa um programa que leia um número n e imprima n linhas no formato a seguir ($n = 6$, no exemplo):

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

23. Construa um programa que leia um número n e imprima n linhas no formato a seguir ($n = 6$, no exemplo):

```
- * * * * *
* - * * * *
* * - * * *
* * * - * *
* * * * - *
* * * * * -
```

PARTE VIII – VETORES NUMÉRICOS

24. Escrever uma função que receba um vetor com 10 valores e retorne quantos destes valores são negativos.

25. Implemente uma função que retorne o maior elemento de um vetor de inteiros de tamanho 10.

26. Implemente uma função que, dado um valor, retorne **1 (UM)** se esse valor pertence a um vetor de inteiros de tamanho 10. Caso não pertença, retorne **0 (ZERO)**.

27. Implemente uma função que retorne a média dos valores armazenados em um vetor de inteiros de tamanho 10.

PARTE IX – VETORES DE CARACTERES

28. Escreva uma função **int contc(char str[], char c)** que retorna o número de vezes que o caracter **c** aparece na string **str**, ambos passados como parâmetros.

```
char texto[]="EXEMPLO";
x=contc(texto,'E'); /* x recebe 2 */
x=contc(texto,'L'); /* x recebe 1 */
x=contc(texto,'W'); /* x recebe 0 */
```

29. Escrever uma função **int ultima(char string[], char c)** que retorna qual a última posição na string em que aparece o caracter **c**. Se o caracter não estiver na string, retornar **-1**.

```
char str[]="teste";
int q;
q=ultima(str, 't'); /* q recebe 3 */
q=ultima(str, 'x'); /* q recebe -1 */
```

30. Escrever uma função **int contabranco(char string[])**, que retorna o número de espaços em branco contidos na string passada como parâmetro.

```
n = contabranco(" a b c"); /* n recebe 3 */
n = contabranco(" abc "); /* n recebe 2 */
n = contabranco("abc"); /* n recebe 0 */
```

31. Escrever um procedimento **void copiaate(char destino[], char origem[], char parar)** que copia para a string **destino** os caracteres da string **origem** que estão antes da primeira ocorrência do caracter **parar** ou até o final de origem, se parar não for encontrado.

```
char str[80];
copiaate(str, "testando a funcao", 'a'); /* str recebe "test" */
copiaate(str, "testando a funcao", 'n'); /* str recebe "testa" */
copiaate(str, "testando a funcao", 'o'); /* str recebe "testand" */
```

PARTE X – MATRIZES

32. Faça uma função que recebe, por parâmetro, uma matriz A(5,5) e retorna a soma dos seus elementos.
33. Faça uma função que recebe, por parâmetro, uma matriz A(7,6) e retorna a soma dos elementos da linha 5 e da coluna 3.
34. Faça uma função que recebe, por parâmetro, uma matriz A(6,6) e retorna o menor elemento da sua diagonal principal.
35. Escreva um algoritmo que leia uma matriz M (5x5). Em seguida calcule e imprima as somas:
- da linha 4 de M
 - da coluna 2 de M
 - da diagonal principal
 - da diagonal secundária.

PARTE XI – PONTEIROS E ALOCAÇÃO DINÂMICA DE MEMÓRIA

36. Faça um programa para alocar espaço em memória para 10 inteiros e peça ao usuário para digitar 10 valores. Posteriormente, imprima os seus respectivos endereço de memórias e o seu conteúdo.
37. Faça um programa que leia uma sequência de 5 números reais do teclado usando ponteiros em lugar de vetores estáticos. Determine o maior e o menor valor.

38. Faça um programa que leia n inteiros (definidos pelo usuário) armazenando-os em uma memória alocada dinamicamente. Em seguida, imprima na tela a quantidade dos n números que são pares e a quantidade que são ímpares.
39. Escreva um programa que lê primeiros os 6 números gerados pela loteria na noite de sábado na TV e depois lê os 6 números jogados por um apostador. Então, o programa compara quantos números o jogador acertou. Em seguida, ele aloca espaço para um vetor de tamanho igual a quantidade de números corretos e guarda os números corretos nesse vetor. Finalmente, o programa exibe os números sorteados (6 números) e os números que o apostador acertou.

PARTE XII – ESTRUTURAS

40. Defina uma estrutura para representar o peso e a altura de uma pessoa. Considere a inicialização de variáveis abaixo:

```
    pessoa joao, maria;  
    joao.altura = 1.90; joao.peso = 98;  
    maria.altura = 1.50; maria.peso = 55;
```

- a) Escreva uma instrução que atribua 1.78 à altura de joao.
 - b) Escreva uma instrução que atribua 75 ao peso de maria.
 - c) Escreva um conjunto de instruções para imprimir a média das alturas e a média dos pesos de joao e maria.
41. Crie uma estrutura chamada ponto contendo apenas as coordenadas x e y (inteiros) do ponto. Declare 2 pontos, leia as coordenadas x e y de cada um e calcule a distância entre eles. Apresente no final a distância entre os dois pontos.
42. Faça um programa que armazene as informações de 11 jogadores de um time de futebol. Cada jogador é identificado pelo número da sua camisa, pelo seu peso, altura e pela inicial do seu nome. O programa deverá ler todas as informações e imprimir ao final a lista completa de jogadores com todas as informações lidas, a inicial do jogador mais baixo e o número do mais pesado.
43. Faça um programa para fazer o cadastro dos veículos de uma empresa. Poderão ser cadastrados no máximo 100 veículos e as informações a serem armazenadas são as seguintes: nome do condutor, placa do veículo e cor do veículo. Armazene também o turno que os veículos são armazenados como sendo m (manhã), t (tarde), n (noite) e i (dia inteiro). Informe ao final a lista completa dos veículos cadastro com todas as informações que forem cadastradas. Utilize procedimentos para fazer a leitura e a escrita dos dados.