

## Lista 2

Prof. Mateus Ferreira Satler

DECSI – ICEA - UFOP

### PARTE I – VETORES NUMÉRICOS

1. Escrever uma função que receba um vetor com 10 valores e retorne quantos destes valores são negativos.
2. Implemente uma função que retorne o maior elemento de um vetor de inteiros de tamanho 10.
3. 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)**.
4. Implemente uma função que retorne a média dos valores armazenados em um vetor de inteiros de tamanho 10.
5. Escrever uma função **int so\_positivo(int vetor[], int tamanho)**, que substitui por zero todos os números negativos do vetor passado por parâmetro, sendo que o número de elementos do vetor é passado para a função no parâmetro **tamanho**. A função deve retornar o número de valores que foram substituídos. Exemplo:

```
int v[5] = {3, -5, 2, -1, 4};  
tr = so_positivo(v,5);  
printf("%d", tr); /* 2 */
```

### PARTE II – VETORES DE CARACTERES

6. 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.
7. Escrever um procedimento **void stringup(char destino[], char origem[])**, que copia todos os caracteres da string **origem** para **destino**, convertendo-os para maiúscula.

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

8. 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 */
```

9. 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 */
```

10. 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 III – MATRIZES

11. Faça uma função que recebe, por parâmetro, uma matriz A(5,5) e retorna a soma dos seus elementos.
12. 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.
13. Faça uma função que recebe, por parâmetro, uma matriz A(6,6) e retorna o menor elemento da sua diagonal principal.
14. Uma matriz quadrada de inteiros é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna, a soma dos elementos da diagonal principal e a soma dos elementos da diagonal secundária são todos iguais. A matriz abaixo é um exemplo de quadrado mágico:
- |    |   |   |
|----|---|---|
| 3  | 4 | 8 |
| 10 | 5 | 0 |
| 2  | 6 | 7 |
- Faça um programa que leia uma matriz quadrada e determine se ela é um quadrado mágico. Utilize funções sempre que possível.

15. 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 IV – RECURSÃO

16. Escreva uma função recursiva que procure um valor em um vetor e retorne o índice do elemento, caso ele exista no vetor, ou -1 caso, caso contrário.
17. Escreva um procedimento recursivo que receba uma string como parâmetro e a exiba invertida.
18. Escreva um procedimento recursivo que imprima os n primeiros elementos da série de Fibonacci. O número n deve ser informado pelo usuário.
19. Faça uma função recursiva que permita inverter um número inteiro N. Ex: 123 – 321
20. Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.