



| INSTRUÇÃO PRÁTICA   |                             | PI-P011 |
|---|-----------------------------|---------|
| MÓDULO  | PI - PROGRAMAÇÃO IMPERATIVA |         |
| OBJETIVO DA ATIVIDADE   |                             | TEMPO   |
| Sedimentar os conceitos de modularização, funções, passagem de parâmetro por valor e por referência, e qualidade na modularização.  |                             | 2h      |
| DESCRIÇÃO   |                             |         |
| <p><b>Exercício 1:</b><br/>Escreva uma função chamada <b>maxmin</b> que receba um vetor de números inteiros chamado <i>vetor</i>, uma variável inteira <i>n</i> contendo o tamanho do vetor, e os endereços de duas variáveis inteiras, <i>maximo</i> e <i>minimo</i> nas quais será retornado o valor do elemento de maior valor e o valor do elemento de menor valor.</p> <p><b>Protótipo da Função:</b></p> <pre>void maxmin(int vetor[], int n, int &amp;maximo, int &amp;minimo);</pre> <p>Escreva também uma função principal (main) que use a função <i>maxmin</i>.</p> <p><b>Exercício 2:</b><br/>Qual o tipo de coesão e acoplamento da função do exercício 1?</p> <p><b>Exercício 3:</b><br/>Faça uma função que receba (por referência) 4 variáveis float e ordena (crescente) os valores destas variáveis. Depois faça o programa principal para testar a sua função.</p> <p><b>Exercício 4:</b><br/>Escreva uma função calcula que:</p> <ol style="list-style-type: none"><li>receba como parâmetros duas variáveis inteiras, X e Y;</li><li>retorne em X a soma de X e Y;</li><li>retorne em Y a subtração de X e Y.</li></ol> <p><b>Exercício 5:</b><br/>Faça uma função "<i>insere_meio(int vet[ ], int tam)</i>" que insere um elemento no meio de um vetor de números inteiros. Depois faça o programa principal para testar a sua função.</p> <p><b>Por exemplo:</b><br/>Sendo o <b>vetor</b> = {1,2,3,4,5,6} e <b>qtde</b> = 6, ao usar<br/><b>insere_meio(vetor, qtde, 100),</b><br/>será retornado o novo valor de <b>qtde</b>, que passou a ser 7, e o vetor, ao término da função, passará a ficar da seguinte forma: <b>vetor</b>= {1,2,3,100,4,5,6}.</p> <p><b>Protótipo da função:</b> <i>int insere_meio(int *vetor, int qtde, int elemento);</i></p> |                             |         |



### Exercício 6:

Faça uma função que receba dois vetores de números inteiros e os seus respectivos tamanhos por parâmetro e retorne um vetor com os **valores dos dois vetores intercalados**. Depois faça o programa principal para testar a sua função.

**Protótipo da função:** `int* intercala(int *vet1, int tam1, int *vet2, int tam2);`

### Exercício 7:

Faça uma função que receba um vetor de números inteiros por parâmetro, a quantidade de elementos do vetor, e um multiplicador e multiplique cada elemento do vetor pelo multiplicador.

**Protótipo da função:** `void multiplica_por_n(int *vet, int qtde, int n);`

### Exercício 8: Instalando Extensões

Faça uma função chamada `conta_primos` que receba um vetor de números inteiros e retorne a quantidade de números do vetor que são primos.

Lembrando que um número primo é aquele divisível apenas por 1 e por ele mesmo.

**Protótipo da função:**

`int conta_primos (int *vet, int qtde);`

### Exercício 9:

Faça uma função para calcular o valor de S, dado por:

$$s = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \dots + \frac{N-1}{2} + \frac{N}{1}$$

**Protótipo da função:** `float calc_serie(int N);`

### Exercício 10:

Escreva uma função que recebe uma string de caracteres e uma letra e devolve um vetor de inteiros contendo as posições (índices no vetor da string) onde a letra foi encontrada) e um inteiro contendo o tamanho do vetor criado (total de letras iguais encontradas). Utilize o retorno de um vetor para retornar os índices e um ponteiro para guardar o tamanho do vetor.

### Exercício 11:

Escreva uma função que codifica uma string em um código secreto. A regra secreta de codificação é extremamente simples: substitui cada letra pela letra seguinte (Z é



codificado como A). Por exemplo, “Estruturas de Dados” se transformaria em “Ftusvuvst ef Ebep”. Escreva uma função para codificar e uma para decodificar cadeias segundo este código. Suas funções devem escrever a string produzida em uma string diferente da fornecida como entrada.

### Exercício 12:

Escreva uma função que receba duas strings A e B por parâmetro e retorne uma terceira string C com os caracteres que aparecem tanto em A quanto em B. O tamanho das string A e B pode ser diferente.

### REFERÊNCIAS

FARRER, Harry et al. **Algoritmos Estruturados**. 3 ed. Rio de Janeiro: LTC, 1999. 260 p.

SCHILDT, Herbert. **C Completo e Total**. São Paulo: Makron, McGraw-Hill, 1990. 889 p

GUIMARÃES, A. de M; LAGES, N. A. de C. **Algoritmos e Estruturas de Dados**. L.T.C, 1994. 216 p.

TREMBLAY, Jean-Paul; BUNT, R.B. **Ciência dos Computadores: uma Abordagem Algorítmica**. Markon, 1997. 384 p.

\_SALVETTI, D. D.; BARBOSA, L. M. **Algoritmos**. Markon, 1997. 274 pg.