

## LISTA DE EXERCÍCIOS 03

### Parte 1: Algoritmos de Classificação e Busca.

1. Dados dois vetores de tamanho 30 contendo números reais, suponha estes estarem ordenados e que não haja nenhum número repetido. Implementar um algoritmo capaz de, a partir dos vetores dados, montar um vetor S contendo todos os elementos dos vetores originais, de forma que S também esteja ordenado.  
Dica: pesquise os algoritmos de classificação por intercalação.  
  
Variação: Generalizar o exercício anterior de modo que o algoritmo possa tratar agora de vetores com valores possivelmente repetidos, embora já ordenados.
2. Implementar o algoritmo de classificação de inserção.
3. Implementar um algoritmo para ordenar um vetor de N números reais pelo método de seleção. Dica: pesquise nos livros indicados.
4. Implementar um algoritmo capaz de cadastrar 50 pessoas através de seus nomes. Uma vez cadastradas, deve-se poder pesquisar um nome dentre os cadastrados através do método de **Busca Binária**, informando o número de cadastro (posição) se encontrado ou a mensagem "Não encontrado" caso contrário.  
Dica: use um vetor de strings para cadastrar os nomes, lembrando-se de mantê-los sempre ordenados. Lembre-se de classificar (ou ordenar) os dados **antes** de tentar procurá-los.
5. Tremblay, Jean Paul; "Ciência dos Computadores"  
(OBS.: faça uma versão dos algoritmos em C, testando as respostas)
  - a) pg 151 - implementar o algoritmo de intercalação simples.  
Use vetores de inteiros.  
Faça uma versão usando vetores de strings.
  - b) pg 184 - ex 5

### Parte 2: Procedimentos e funções.

1. Escreva um subprograma capaz de calcular a média aritmética de três parâmetros passados.
2. Escreva um subprograma capaz de calcular a área de qualquer figura geométrica do tipo circunferência. Lembre-se:
$$Area_{CIRC} = \pi \times Raio^2$$
3. Escreva um subprograma que execute a operação de potenciação expressa por  $x^n = e^{n \cdot \ln(x)}$  (você sabe por que isso é válido? Pergunte !!!)
4. Escreva um subprograma capaz de trocar o conteúdo de 2 variáveis inteiras passadas como parâmetro. Dica: use passagem de parâmetros POR REFERÊNCIA.
5. Modifique o exercício que calcula a média de três números para que seja calculada a média aritmética de um vetor inteiro passado como parâmetro. Dica: passe também como parâmetro a quantidade de elementos a serem considerados.
6. Faça um subprograma que calcule o seno de um número passado como parâmetro. Use o algoritmo representado no exercício 1 dos desafios da Lista de Exercícios 2.
7. Escrever um subprograma que receba como parâmetro uma string de até 80 caracteres e a devolva ordenada em ordem ascendente. Use passagem de parâmetro por referência.

8. Escreva um subprograma que, recebendo duas strings como parâmetro, seja capaz de verificar se a primeira existe dentro da segunda.
9. Números palíndromos são aqueles que escritos da direita para a esquerda têm o mesmo valor. Ex. 929, 545, 97379. Escrever um subprograma que, recebendo um número inteiro como parâmetro, retorne o mesmo escrito ao contrário.
10. Usando o subprograma que calcula números invertidos, escrever um programa que calcule todos os palíndromos entre 1 e 1000.
11. Escrever um subprograma que retorne se uma palavra existe ou não dentro de um vetor de 1000 posições de strings de tamanho 30. Aplique o método de busca binária, considerando que este vetor está ordenado pelo valor das strings.
12. Escrever um subprograma que ordene o vetor de strings do exercício número 7. Use qualquer algoritmo que você tenha conhecimento ("Bubble", Seleção, etc.).
13. Escrever um subprograma que, recebendo como parâmetros uma matriz (por referência) e um inteiro (por valor), multiplique a matriz pelo inteiro.
14. Escrever um subprograma capaz de multiplicar duas matrizes A e B, passadas por referência.
15. Escreva um subprograma que devolva o valor numérico 1 (um) caso uma matriz passada como parâmetro seja simétrica e 0 (zero) caso contrário. Os parâmetros devem incluir as dimensões da matriz.
16. escreva um subprograma capaz de calcular a área de uma figura geométrica do tipo Circunferência, Triângulo ou uma figura de 4 lados (quadrado, trapézio, paralelogramo). As fórmulas para cálculo das áreas são as que seguem:

$$Area_{CIRC} = \pi \times Raio^2$$

$$Area_{TRIANG} = \frac{Base \times Altura}{2}$$

$$Area_{PARALELOG} = \frac{(BASE + base) \times Altura}{2}$$

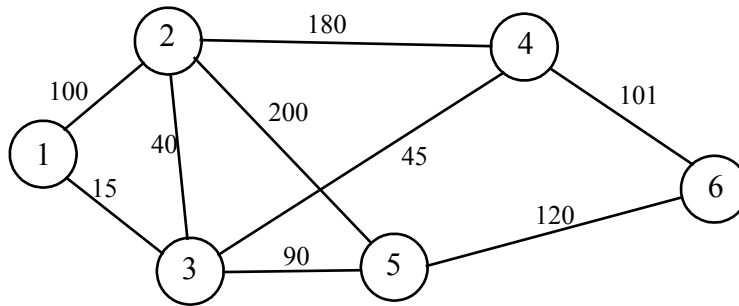
17. Tremblay, Jean Paul; "Ciência dos Computadores"

- a) pág 238 - ex 1, 7, 4
- b) pág 249 - ex 3, 4, 5, 6
- c) pág 269 - ex 2, 3

-----  
**\*- DESAFIOS \*-**  
 -----

1. Implemente o algoritmo do QuickSort, que usa subprogramas e recursão. Compare-o aos outros algoritmos que você conhece de ordenação ("Bubble-sort" ou bolha, seleção, intercalação) em relação ao número de operações de troca efetuadas.
2. Implemente o algoritmo de classificação por Intercalação Múltipla (Tremblay, pág. 155) usando os conceitos de subprogramas.
3. Implemente o algoritmo de Diagramação (Tremblay, pág 219).
4. Altere o exercício 1 de forma que este opere com strings.

5. Os nós do grafo, apresentado abaixo, representam cidades, e os arcos, a presença de uma estrada ligando estas duas cidades. Os números ao lado dos arcos representam a distância medida em quilômetros:



Pode-se representar este grafo através de uma variável composta bidimensional (uma matriz)  $D$ , na qual a existência de conexão entre duas cidades  $i, j$  é indicada pelo elemento  $D[i, j]$  diferente de 0. Assim:

	1	2	3	4	5	6
1	0	100	15	0	0	0
2	100	0	40	180	200	0
3	15	40	0	45	90	0
4	0	180	45	0	0	101
5	0	200	90	0	0	120
6	0	0	0	101	120	0

Matriz de Distâncias  $D$

O problema todo consiste, então, em se descobrir qual o menor caminho entre duas cidades quaisquer. Este problema foi resolvido pelo matemático Dijkstra em 1971 e tem aplicações marcantes em diversos problemas de otimização, como planejamento de voo, linhas de ônibus, percursos de trens de carga, colocação de semáforos etc. Pesquise e implemente o algoritmo de Dijkstra para a solução deste problema.