



## INSTITUTO DE ENGENHARIAS E CIÊNCIAS DO MAR

### Ficha de Trabalho #2

▷ **Tópicos:**

- Vetores
  - Estruturas
  - Ponteiros;
  - Funções:
    - Passagem por valor
    - Passagem por referência
  - Alocação
1. Escreva uma função **hm** que converta minutos em horas-e-minutos. A função recebe um inteiro **t** e os endereços de duas variáveis inteiras, digamos **h** em, e atribui valores não-negativos a essas variáveis de modo que tenhamos  $m < 60$  e  $60h + m = t$ . Escreva também um programa que use a função **hm**.
  2. Escreva uma função **mm** que receba um vetor inteiro  $v[0...n-1]$  e os endereços de duas variáveis inteiras, digamos **max** e **min**, e guarde nestas variáveis o valor de elemento mínimo e o valor de um elemento máximo do vetor. Escreva um programa também um programa que use a função **mm**.
  3. Escreva um programa que leia um inteiro positivo **n** seguido de **n** números inteiros e imprima esses **n** números em ordem invertida. O programa não deve impor quaisquer restrições ao valor de **n**.
  4. Um treinador de atletismo treina **n** atletas e faz **m** sessões de treino por semana. Em cada sessão, cada atleta percorre uma distância que é cronometrada. Os valores dos tempos, em segundos, são registados sob a forma de matriz, onde cada linha identifica um atleta e cada coluna uma sessão de treino. O número de atletas e de sessões só são conhecidas durante a execução do programa. Escreva um programa:
    - a) calcular e escrever a média dos tempos realizados em cada sessão de treinos;
    - b) determinar e escrever o melhor tempo realizado por cada um dos atletas durante as sessões.

Neste exercício, implemente os seguintes subprogramas:

Para ler os tempos: void leMatriz(float tempos[ ][ ],int atletas,int sessoes);

Para calcular e escrever a média dos tempos em cada sessão de treinos: void escreveMediaColunas(float tempos[ ][ ], int atletas, int sessoes);  
Para calcular e escrever o melhor tempo realizado por cada um dos atletas: void escreveMinimoLinhas(float tempos[ ][ ], int atletas, int sessoes);

5. Números complexos são identificados pela sua parte real e imaginária. Faça um programa que lê dois números complexos (um de cada vez) no formato  $x+yi$  (ex:  $5+3i$ );
6. Implemente uma função complexo soma(complexo a, complexo b) que recebe dois números complexos como argumento e devolva a soma dos dois;
7. Crie um novo tipo de dados chamado DVD para guardar informação sobre filmes disponíveis num videoclube. Cada DVD deve conter campos para o título, duração (em minutos) e um campo que indica se está ou não disponível para aluguer.
8. Use o tipo definido no exercício anterior para criar um vector que representa a oferta de um videoclube. Escreva um programa que leia do teclado informação para preencher 3 registos de DVD e mostre no ecrã:
  - a) o título do filme mais curto;
  - b) a duração do filme mais longo;
  - c) os títulos dos filmes disponíveis para aluguer;
  - d) todo o conteúdo do vector de DVDs.
9. Implemente duas versões da função **strlen**, uma em que utiliza índices e outra em que aplique aritmética de ponteiros.
10. Altere a função desenvolvida no exercício 6 por forma a que esta não devolva o resultado da soma mas que o resultado da soma venha no primeiro argumento da função
11. Altere a função desenvolvida no exercício anterior por forma a que a função devolva um ponteiro para um número complexo que tem o resultado da soma.
12. Considere que se pretende ler do standard input a informação de um conjunto de inscrições de alunos a disciplinas. A estrutura que guarda a informação de uma inscrição está definida do seguinte modo:

```
typedef struct {  
    int num_aluno;  
    char* cod_disc;  
} Inscricao;
```

Escreva em linguagem C a função **Inscricao\* lerInscricoes(int n)** que lê um sequência de **n** inscrições do standard input e devolve um novo vector de inscrições de alunos em disciplinas. Assuma que o código de uma disciplina é uma sequência

de letras e dígitos com tamanho máximo de 10 caracteres. Escreva um programa que use a função **Inscricao\* lerInscricoes(int n)**.