



Universidade do Minho
Escola de Engenharia

MIETI :: Métodos de Programação II
2015/16

Práticas Laboratoriais

Módulo 3

António Esteves
esteves@di.uminho.pt
Março 2016

Exercício 1 - Arrays e funções com argumentos que são apontadores

- ◇ Escrever um programa em C que declara, dentro do `main()`, um *array* de inteiros com nome `val` e 10 posições, inicializado na declaração com valores à escolha.
- ◇ Implemente uma função `soma()` que soma os valores do *array* `val` e devolve o resultado.
- ◇ Chamar a função `soma()` no `main()` e escrever o resultado da soma.

Exercício 2 - Estruturas

- ◇ Escrever um programa em C que utiliza uma estrutura `struct ponto` contendo 2 campos inteiros `x` e `y`. A função `main()` deve:
- declarar duas variáveis `p0` e `p1` do tipo `struct ponto`
 - inicializar `p0` com `x=24` e `y=35`
 - inicializar `p1` com `x=62` e `y=58`
 - calcular a distância euclidiana `d` entre `p0` e `p1` (usando a fórmula apresentada em baixo)
 - escrever o valor de `d` no ecrã

$$d = \sqrt{(x1 - x0)^2 + (y1 - y0)^2}$$

Exercício 3 - Estruturas e funções com parâmetros que são apontadores para estruturas

- ◇ Escrever um programa em C que utiliza uma estrutura `struct ponto` contendo 2 campos inteiros `x` e `y`. A função `main()` deve:
 - declarar duas variáveis `p0` e `p1` do tipo `struct ponto`
 - inicializar `p0` com `x=10` e `y=20`
 - inicializar `p1` com `x=15` e `y=27`
 - chamar a função `distanciaEuclidiana()`, usando como argumentos o endereço da variável `p0` e o endereço de `p1`
 - escrever o valor devolvido pela função no ecrã.
- ◇ Escrever a função `distanciaEuclidiana()` que tem dois argumentos (i) um apontador para um ponto `pa` e (ii) um apontador para outro ponto `pb`. A função calcula a distância euclidiana `d` entre `pa` e `pb` e devolve o valor de `d`.

Exercício 4 - Alocação dinâmica de memória para um array

- ◇ Escrever um programa em C que declara um apontador `ptr` para `double`'s
 - (1) Escrever `5.75` na posição afastada 100000 posições do início da zona apontada por `ptr`
 - (2) Ler o conteúdo da posição escrita anteriormente
 - (3) Escrever o valor lido no ecrã
- ◇ Acha que estes 3 passos podem ser executados sempre com sucesso?
- ◇ Se acha que não, altere o programa de modo a reservar espaço em memória para (um *array* de) 100010 `double`'s, usando a função `malloc()`, e colocar `ptr` a apontar para o início desse espaço

Em que local se insere este passo na sequência de tarefas anteriores?

- ◇ No fim do programa libertar a memória reservada

Exercício 5 - Programa com argumentos

- ◇ Escrever um programa em C que aceita 2 argumentos quando é executado
- ◇ Pretende-se que os 2 argumentos aceites pelo programa sejam 2 valores inteiros
- ◇ O programa compara esses 2 valores e indica se são iguais, se o primeiro é maior ou se o segundo é maior

Exercício 6 - Acesso a ficheiros e array de estruturas

- ◇ Escrever um programa em C que declara um *array* de estruturas do tipo `struct cambio` no `main()`, com dimensão 5. A estrutura contém 2 campos:
 - `moeda` (*string* com 20 caracteres para guardar o nome duma moeda)
 - `cotacaoEUR` (`float` com a cotação da `moeda` em relação ao EURO).
- ◇ Preencha o *array* com o nome de 5 moedas e respetiva cotação
- ◇ Escreva os valores da estrutura num ficheiro texto `ex6.txt`, com o seguinte formato por linha:

`nomeMoeda cotacaoMoeda`