

# Programação I

## Exercícios 6 - Alocação dinâmica de memória

```
#include <stdlib.h>
```

```
void *malloc(size_t size);  
void *calloc(size_t number, size_t size);  
void *realloc(void *ptr, size_t size);  
void free(void *ptr);
```

### Exercício 1:

Implementar um programa que concatene 2 strings. O programa deverá pedir ao utilizador a 1ª e a 2ª string e deverá apresentar para o ecrã as duas strings concatenadas. As strings de entrada não têm mais que 99 caracteres. A string final deverá **ocupar apenas a memória estritamente necessária**.

### Exercício 2:

Escreva um programa para calcular a média de  $n$  números e imprima para o ecrã os números inferiores a metade da sua média. O número  $n$  é um inteiro introduzido pelo utilizador, assim como os  $n$  números. O programa não deve impor limitações sobre o valor de  $n$  e deve reservar a memória estritamente necessária.

Exemplo de execução:

Introduza o nº de números que pretende introduzir:

5

Introduza os 5 números:

10

4

5

25

20

Números inferiores a (12.8): 10,4,5

### Exercício 3:

Implementar um programa que vá aceitando valores do utilizador e os guarde num vector, **alocado dinamicamente**. A leitura termina quando o utilizador introduzir o valor 0 (zero). Após a leitura, o programa deve apresentar a soma dos valores do vector.

**Exercício 4:**

Escreva um programa que leia um número inteiro positivo  $n$  e  $n$  números inteiros e imprima esses  $n$  números por ordem inversa à ordem lida do teclado. O programa não deve impor limitações sobre o valor de  $n$  e deve reservar o espaço de memória estritamente necessário.

Exemplo de execução:

Introduza o número de números: 3

Introduza os números

2

989

454

O programa deve imprimir: 454 989 2.

**Exercício 5:**

Escreva um programa que leia um ficheiro cujo nome é recebido como argumento de linha de comando. O ficheiro é constituído por um valor inteiro por linha. A primeira linha do ficheiro indica o número de linhas que constituem o resto do ficheiro. O programa deverá escrever para o ecrã os valores que são inferiores ou iguais ao último valor lido do ficheiro.

Exemplo de ficheiro:

7

45

67

134

220

23

456

80

O programa deverá escrever para o ecrã os valores: 45, 67, 23 e 80

**Exercício 6:**

Desenvolva um programa que receba do utilizador um número de valores inteiros a introduzir posteriormente. De seguida, deve receber os números, ordená-los por ordem crescente e imprimir para o ecrã os números ordenados. Os números introduzidos devem ocupar a memória estritamente necessária e a memória deve ser libertada antes do término do programa.

**Exercício 7:**

Pretende-se implementar um programa que receba do utilizador um número, o nome e o preço dos produtos comercializados numa loja de produtos tradicionais. Pretende-se listar o nome dos produtos com preço acima da média. O programa deve reservar o espaço de memória estritamente necessário, por isso, deve pedir ao utilizador para identificar o número de produto comercializados pela loja. O programa deve libertar a memória antes do término do programa.