

## Aula prática 2

Estes exercícios têm como objetivo aplicar conhecimentos de alocação dinâmica de memória apresentados ainda em Programação 1 e na segunda aula teórica de Programação 2. Adicionalmente, será feita uma aplicação prática destes conceitos, utilizando uma biblioteca de manipulação de vetores de *strings*.

1 – Escreva um programa que leia, para um vetor V, um número N (escolhido pelo utilizador) de elementos inteiros. Utilizando funções, construa e exiba dois novos vetores baseados no vetor V: um com os seus números pares e outro com os seus números ímpares (ver exemplo). Todos os vetores deverão ser criados dinamicamente.

### Exemplo

```
Número de elementos? 5
Introduza o 1º elemento: 1
Introduza o 2º elemento: 0
Introduza o 3º elemento: 3
Introduza o 4º elemento: 8
Introduza o 5º elemento: 10

Vetor original: [ 1 0 3 8 10 ]
Números pares : [ 0 8 10 ]
Números ímpares: [ 1 3 ]
```

2 – Pretende-se preencher um vetor de 3 *strings* lidas do teclado, cada uma com um máximo de 80 caracteres. Para não desperdiçar espaço em memória, cada *string* deve ser guardada no vetor, alocando dinamicamente apenas o número de caracteres necessário.

O programa deverá permitir que o utilizador escolha em que posição (de 1 a 3) do vetor a *string* é armazenada. Sempre que o utilizador inserir uma *string* para uma posição já utilizada, a memória alocada para a *string* anterior deve ser redimensionada usando a função *realloc()*. O programa deve ser robusto relativamente à inserção de posições inválidas (ver exemplo). O programa termina quando for indicada a posição 0.

### Exemplo

```
[1] <vazio>
[2] <vazio>
[3] <vazio>
Posição para a nova string: 2
Nova string: primeira frase

[1] <vazio>
[2] primeira frase
[3] <vazio>
Posição para a nova string: 11
Posição inválida. Posição para a nova string: 1
Nova string: segunda frase

[1] segunda frase
[2] primeira frase
[3] <vazio>
Posição para a nova string: 2
Nova string: TERCEIRA frase

[1] segunda frase
[2] TERCEIRA frase
[3] <vazio>
Posição para a nova string: 0
```

3 – O ficheiro `vetor.zip` contém uma biblioteca implementada em C para a manipulação de vetores de *strings*. Esta biblioteca é composta por diversas funções que podem ser usadas, por exemplo, para criar um novo vetor, apagar um vetor já existente, adicionar elementos, eliminar elementos, ordenar os elementos, etc.

3.1 – Estude a implementação da biblioteca fornecida.

3.2 – Crie um pequeno programa de teste da biblioteca utilizando o IDE Eclipse. Se não souber como criar um projeto em Eclipse, consulte o breve tutorial sobre Eclipse que está disponível no Moodle.

O programa de teste deverá realizar as seguintes operações:

1. Criar um novo vetor vazio.
2. Solicitar ao utilizador 5 *strings* para inserir no vetor.
3. Imprimir o conteúdo do vetor.
4. Solicitar ao utilizador uma *string*; se essa *string* existir no vetor, apagar a *string*.
5. Imprimir o conteúdo do vetor.
6. Ordenar o vetor.
7. Imprimir o conteúdo do vetor.

3.3 – Implemente um programa semelhante ao do exercício 2 utilizando agora a biblioteca de manipulação de vetores de *strings*.

3.4 – Adicione as seguintes funções à biblioteca:

**vetor \* vetor\_concatena (vetor \*vec1, vetor \*vec2)**

*cria um novo vetor que resulta da concatenação de dois outros vetores*

**int vetor\_inverte (vetor \*vec)**

*inverte os elementos do vetor*

**vetor\* vetor\_baralha (vetor \*vec)**

*cria um novo vetor com os mesmos elementos do vetor vec mas guardados em posições aleatórias*

Altere o programa de teste de forma a verificar também estas funções.