

Ficha EXTRA: Vetores, matrizes e *strings*

1. Elabore um programa que crie um vetor de 10 posições e em seguida peça ao utilizador dois valores X e Y quaisquer, correspondentes a duas posições no vetor. No final o programa deverá escrever a soma dos valores encontrados nas respetivas posições X e Y.

Exemplo:

10	20	30	40	50	60	70	80	90	100
0	1	2	3	4	5	6	7	8	9

Posição X: 1
Posição Y: 9

A soma do valor das posições é: 120

Sugestão: inicie o vetor com recurso a uma estrutura de repetição.

2. Elabore um programa que auxilie na gestão da ocupação dos 10 quartos de uma determinada pensão. O menu principal deverá ser o seguinte:

```
(R)eservar  
  
(D)esmarcar reserva / desocupar  
  
(V)er estado de ocupações  
  
(S)air  
  
OPCAO ->
```

(R)eservar

Esta opção apresenta o estado das ocupações e solicita ao utilizador o quarto a reservar.

```
0 -> Ocupado
1 -> Livre
2 -> Livre
3 -> Ocupado
4 -> Livre
5 -> Ocupado
6 -> Livre
7 -> Livre
8 -> Livre
```

(D)esmarcar reserva / desocupar

Esta opção apresenta o estado das ocupações e solicitar ao utilizador o quarto a desmarcar.

(V)er estado de ocupações

Nesta opção deverá ser mostrado o estado de ocupações.

Na implementação do programa deverão ser elaboradas as seguintes funções:

- **menuPrincipal()**: apresenta o menu principal, e obtém a opção selecionada pelo utilizador.
- **inicializarQuartos()**: coloca todos os quartos como livres. (sugestão: o vetor poderá ser de inteiros onde 0 corresponde a livre e 1 corresponde a ocupado).
- **estadoQuartos()**: indica o estado (livre ou ocupado) de todos os quartos.
- **quartosNumEstado()**: contabiliza o número de quartos que se encontram livres ou ocupados (conforme o estado que o utilizador pretender analisar).
- **leQuarto()**: solicita ao utilizador o número do quarto que pretende reservar ou desmarcar. Deverá também validar o número do quarto e assegurar que o quarto se encontra livre se se pretender reservar ou ocupado se se pretender desmarcar.

3. Elabore um programa que armazene as notas dos 26 alunos inscritos em Fundamentos de Programação. O programa deverá:

- Pedir ao utilizador as notas dos alunos
- Calcular e indicar a média das notas
- Indicar a nota mais alta
- Indicar a nota mais baixa
- Escrever no ecrã todas as notas positivas
- Escrever no ecrã todas as notas negativas

Na implementação deste programa deve elaborar as seguintes funções:

ler_num: deverá ler do teclado um número entre -1 e 20.

ler_notas: deverá pedir ao utilizador as notas respeitantes aos alunos. Esta leitura deve terminar quando já foram introduzidas todas as notas ou quando o utilizador digitar -1. Use a função **ler_num** criada anteriormente.

calcula_media: calcula e devolve a média das notas dos alunos.

nota_superior: procura e devolve a nota mais alta.

nota_inferior: procura e devolve a nota mais baixa.

notas_positivas: escreve no ecrã todas as notas superiores ou iguais a 10.

notas_negativas: escreve no ecrã todas as notas inferiores a 10.

4. Elabore um programa que crie uma matriz 10 x 10 e leia um valor X. O programa deverá imprimir no ecrã a matriz, bem como fazer uma pesquisa do valor X na matriz e escrever a localização (linha e coluna) ou uma mensagem de “não encontrado”.

Sugestão: inicialize a matriz com recurso a uma estrutura de repetição.

5. É dada a seguinte tabela em cujas colunas figuram os modelos automóveis de determinada marca, cilindrada e velocidade máxima, respetivamente da esquerda para a direita:

<i>Modelo</i>	<i>Cilindrada</i>	<i>Vel. Máx.</i>
5001	1100	155
5003	1315	160
5005	1495	180
5007	1660	220
6009	1900	190
6030	1295	175

Elabore um programa que permita consultar a informação a tabela de modo a conseguir saber e imprimir qual a cilindrada de determinado veículo, sabendo o modelo que é perguntado ao utilizador.

6. Resolva as seguintes alíneas:

- Defina uma função que receba uma *string* e devolva o número de vogais que contém.
- Defina uma função que receba uma *string* e converta todas as suas vogais para maiúsculas.
- Defina uma função que receba uma *string* e lhe retire todos os espaços em branco.
- Defina uma função que inverta uma *string* recebida como parâmetro.

- e. Elabore um programa que peça ao utilizador uma *string* à sua escolha e que utilize as funções anteriores, mostrando no monitor o resultado de cada função.
7. Faça um programa que peça ao utilizador o seu nome completo e coloque numa outra *string* as suas iniciais (em maiúsculas, mesmo que o utilizador tenha escrito o nome em minúsculas). A cada inicial deverá ser acrescentado um ponto e um espaço. (A última não necessita de espaço).

Exemplo:

Nome completo: Serafim Paços Dias Aguiar de Mota

Iniciais: S. P. D. A. M.

8. Elabore um programa que peça ao utilizador uma *string* e lhe retire todos os espaços que estão a mais. Entendem-se por espaços a mais os que estão no início e no fim da *string* e os que estão entre palavras, sempre que existir mais do que 1 espaço contíguo.
9. Escreva uma função que determine se uma palavra é ou não uma capicua. Esta função não deve fazer a distinção entre maiúsculas e minúsculas. Escreva ainda um pequeno programa para testar a função.