



FEUP

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO  
Mestrado em Engenharia Informática e Computação  
Fundamentos da Programação

Auto Teste  
AT08  
**FACULTATIVO**  
Duração: 60 min.

**NOTAS IMPORTANTES:**

- 1 - Deve respeitar rigorosamente os nomes dos procedimentos que são indicados bem como os formatos de saída dos resultados.
- 2 - Não utilize caracteres acentuados nos nomes dos procedimentos nem nos parâmetros.
- 3 - Utilize comentários só "with Semicolons" ("com ponto e vírgula") e nunca "with a Box" ("com uma caixa").
- 4 - O código desenvolvido durante a prova, contido num único ficheiro com a extensão ".scm", deve ser submetido no Moodle usando o "link" correspondente à prova realizada. A não observação desta regra levará a que o código submetido não possa ser avaliado.
- 5 - Antes de submeter o ficheiro, assegure que este não tem erros de sintaxe (não dá erro ao premir o botão "correr").
- 6 - Assegure que o ficheiro não produz qualquer output (não há resultado visível no ecrã ao premir o botão "correr").

(cotação: 25% + 25% + 50%)

**Exercício 1**

Considere um vetor de nomes de pessoas. Cada nome, uma string, poderá ter mais do que uma palavra (separadas por espaços).

- 1.1 - **Desenvolva** o procedimento **conta-letras** que conta o número de ocorrências de uma determinada letra numa string. Este procedimento tem dois argumentos: a string e o carácter a contar.  
Obs.: A letra deve ser sempre contabilizada independentemente de ser minúscula ou maiúscula.

Considere os seguintes exemplos:

```
> (conta-letras "Antonio Fernando Coelho" #\o)
5
```

```
> (conta-letras "Antonio Fernando Coelho" #\a) ; a letra a aparece como A e como a
2
```

- 1.2 - **Desenvolva** o procedimento **conta-vogais** que conta o número de ocorrências de cada uma das vogais (**a e i o u**) num vetor de strings. Este procedimento recebe como parâmetro um vetor de strings, com os nomes, e retorna um vetor com cinco valores numéricos, que contabilizam o número de ocorrências das 5 vogais (tanto maiúsculas como minúsculas).  
Sugestão: Utilize o procedimento definido na alínea anterior.

Considere o seguinte exemplo:

```
> (define nomes (vetor "Antonio Silva" "Joaquim Oliveira" "Ana Meireles"))
> (conta-vogais nomes)
#5(6 4 6 4 1)
```

**Exercício 2**

No sistema informático de uma loja, as vendas realizadas num dia representam-se por estruturas do tipo "lista de pares":

```
(define todas-as-vendas (list (cons "Manuel" 235) (cons "Maria" 45) (cons "Manuel" 25) (cons "Ana" 457)))
```

O dono da loja prefere os resultados acumulados por vendedor. Para a geração da lista final deve eliminar os elementos repetidos (acumulando os valores das vendas), mas preservando a ordem de ocorrência da 1ª venda de cada vendedor.

**Desenvolva** o procedimento **acumula!**, que modifica a lista de vendas tal como se pode observar no exemplo seguinte:

```
> todas-as-vendas
(("Manuel" 235) ("Maria" 45) ("Manuel" 25) ("Ana" 457))
> (acumula! todas-as-vendas)
> todas-as-vendas
(("Manuel" 260) ("Maria" 45) ("Ana" 457))
```

----- FIM da Prova Prática -----