



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Mestrado em Engenharia Informática e Computação
Fundamentos da Programação - 2010/2011

Prova Prática 1
Duração: 70 min.
10 Nov 2010

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").

Nome: _____ Hora de entrega da prova: _____

Companhias de Aviação

1- (20 valores em 100)

Uma companhia de aviação possui uma frota de aviões composta por vários aparelhos de diferentes modelos. Cada modelo de avião caracteriza-se pelo número máximo de passageiros (lotação máxima) que pode transportar. Pretende-se determinar quantos aviões (de um determinado modelo) são necessários para transportar um determinado número de passageiros.

Desenvolva o procedimento **calcula-avioes** que possui dois parâmetros: o número de **passageiros** a transportar e a **lotação** máxima do modelo de avião a utilizar. Este procedimento devolve o número de aviões a utilizar.

Observe os seguintes exemplos de utilização deste procedimento:

```
> (calcula-avioes 1400 500)
3
> (calcula-avioes 1010 200)
6
> (calcula-avioes 400 200)
2
```

Completar o procedimento **calcula-avioes**:

```
(define calcula-avioes
  (lambda (passageiros lotacao)
    ...
```

2- (20 valores em 100)

Considere agora uma outra companhia aérea, de menor dimensão, que apenas possui três aviões, cada um dos quais possui a lotação indicada:

Airbus **A380** - 500 passageiros
Airbus **A330** - 300 passageiros
Airbus **A320** - 150 passageiros

De forma a poder reservar um aparelho, necessitamos de saber qual o avião mais adequado para transportar um determinado número de passageiros. Deverá ser seleccionado o menor aparelho que tiver capacidade para os transportar ("**A320**", "**A330**" ou "**A380**").

Desenvolva o procedimento **escolhe-aviao** que visualiza um texto com o nome do modelo do aparelho que melhor se adequa a esse transporte. O procedimento tem apenas um parâmetro com o número de **passageiros** a transportar. *Obs.: Caso o número de passageiros seja superior à lotação do maior dos aparelhos (o A380) ou o número de passageiros seja inferior a 1, o procedimento deverá visualizar "Nenhum".*

Observe os seguintes exemplos de utilização deste procedimento:

```
> (escolhe-aviao 256)
A330 ; NOTA: esta mensagem não deve ser seguida de (newline)
> (escolhe-aviao 10)
A320 ; NOTA: esta mensagem não deve ser seguida de (newline)
> (escolhe-aviao 700)
Nenhum ; NOTA: esta mensagem não deve ser seguida de (newline)
```

Completar o procedimento **escolhe-aviao**:

```
(define escolhe-aviao
  (lambda (passageiros)
    ...
```

3- (25 valores em 100)

Considere que as companhias de aviação têm que pagar aos aeroportos uma taxa de 1000€ por cada avião e outra taxa de 50€ por passageiro.

Desenvolva o procedimento **taxa-aeroporto** que devolve o valor total a pagar por uma companhia de aviação para transportar um determinado número de **passageiros** com um determinado modelo de avião (caracterizado pela sua **lotação** máxima). Os parâmetros deste procedimento são o número de **passageiros** e a **lotação** máxima do avião. Considere ainda que a partir do 11º avião, a taxa por avião tem uma redução de 50%.

*Obs.: Sugere-se a utilização do procedimento **calcula-avioes** que desenvolveu na alínea 1.*

Observe os seguintes exemplos de utilização deste procedimento:

```
> (taxa-aeroporto 150 100)
9500      ; cálculo: 150*50 + 2*1000

> (taxa-aeroporto 450 200)
25500     ; cálculo: 450*50 + 3*1000

> (taxa-aeroporto 1250 100)
74000     ; cálculo: 1250*50 + 10*1000 + 3*500 porque são mais de 10 aviões (treze aviões)
```

Completar o procedimento **taxa-aeroporto**:

```
(define taxa-aeroporto
  (lambda (passageiros lotacao)
    ...
```

Trabalhar com dígitos... e com recursividade.

4- (20 valores em 100)

Considere o seguinte procedimento que determina se um determinado **dígito** (número entre 0 e 9) é **múltiplo de três**:

```
(define e-mult3?
  (lambda (digito)
    (cond
      ((or
        (= digito 3)
        (= digito 6)
        (= digito 9)) #t)
      (else #f))))
```

Desenvolva um procedimento, denominado **conta-dígitos-mult3**, que conte o **número de dígitos múltiplos de 3** que um determinado **número** positivo contém. Este procedimento tem um único parâmetro, que é o **número positivo** a analisar e devolve o número de dígitos nessa situação.

*Obs.: Considere que são apenas utilizados **números positivos**. Utilize o procedimento **e-mult3?**.*

Observe os seguintes exemplos de utilização deste procedimento:

```
> (conta-dígitos-mult3 9203)
2

> (conta-dígitos-mult3 1245780)
0

> (conta-dígitos-mult3 369)
3
```

Completar o procedimento **conta-dígitos-mult3**:

```
(define conta-dígitos-mult3
  (lambda (numero)
    ...
```

5 - (15 valores em 100)

Desenvolva o procedimento **cria-numero-mult3** que devolve um número inteiro, composto apenas pelos dígitos múltiplos de 3 que constituem o **número** passado como parâmetro do procedimento, na mesma ordem em que aparecem nesse mesmo número.

*Obs.: Utilize o procedimento **e-mult3?**.*

Observe os seguintes exemplos de utilização deste procedimento:

```
> (cria-numero-mult3 123456)
36

> (cria-numero-mult3 11111)
0

> (cria-numero-mult3 2334567)
336
```

Completar o procedimento **cria-numero-mult3**:

```
(define cria-numero-mult3
  (lambda (numero)
    ...
```

----- FIM da Prova Prática -----