



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO  
Mestrado em Engenharia Informática e Computação  
Fundamentos da Programação - 2012/2013

Prova Prática 1  
Duração: 75 min.  
7 Nov 2012

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-se de 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").
- 7 - Durante a prova apenas poderá aceder às páginas do Moodle de Fundamentos da Programação, com exceção dos fóruns. A prova prática será monitorizada e qualquer acesso a outro recurso do Moodle que não os de Fundamentos da Programação deste ano letivo implicará a anulação da prova.

Nome: \_\_\_\_\_ Hora de entrega: \_\_\_\_\_ Computador: \_\_\_\_\_

Peça de artilharia

Parabéns! Você acabou de ser recrutado para participar na construção da nova peça de artilharia do exército português. Deverá desenvolver um conjunto de procedimentos para o sistema de controlo deste novo canhão.

1- (10 valores em 100)

Desenvolva o procedimento **graus->rad** que converte um ângulo em graus para radianos.

$$\text{radianos} = \frac{\text{graus} \cdot \pi}{180}$$

Obs.: Pode definir pi como (define pi (acos -1))

Completar o procedimento **graus->rad**:

```
(define graus->rad  
  (lambda (graus)  
    ...
```

2- (15 valores em 100)

Desenvolva o procedimento **alcance-projetil** que determina o alcance (em metros) de um projétil lançado por este novo canhão, com uma determinada velocidade inicial (**v<sub>0</sub>**) e um determinado ângulo de elevação (**θ**) em graus. A fórmula para o cálculo é a seguinte, considerando **g = 9.8 m/s<sup>2</sup>**.

$$\text{alcance} = \frac{v_0^2 \cdot \sin(2\theta)}{g}$$

O procedimento **alcance-projetil** possui dois parâmetros: a velocidade inicial **v<sub>0</sub>** e o **ângulo** de elevação. Este procedimento devolve o número correspondente alcance do projétil.

Obs.: Considere a existência do procedimento **graus->rad** (mesmo que não o tenha feito na alínea anterior) para fazer a conversão para radianos, essencial para os argumentos dos procedimentos trigonométricos.

Observe os seguintes exemplos de utilização deste procedimento:

```
> (alcance-projetil 100 30)  
883.699391616774  
> (alcance-projetil 1000 45)  
102040.8163265306  
> (alcance-projetil 100 89)  
35.611731329082346
```

Completar o procedimento **alcance-projetil**:

```
(define alcance-projetil  
  (lambda (v0 angulo)  
    ...
```

3- (20 valores em 100)

Considere agora que pretende desenvolver uma bússola simples com apenas os quatro pontos cardiais, que serão mostrados no ecrã de acordo com o seu azimuth. Para tal deverá utilizar a seguinte tabela:

azimute	ponto cardinal
-45° < azimute <= 45°	N
45° < azimute <= 135°	E
135° < azimute <= 225°	S
225° < azimute <= 315°	O

Obs.: O texto a visualizar é apenas uma cadeia de texto com apenas um carácter maiúsculo.

Observe os seguintes exemplos de utilização deste procedimento:

```
> (bussola 90)
E
> (bussola 180)
S
> (bussola 270)
O
```

Completar o procedimento **bussola**:

```
(define bussola
  (lambda (angulo)
    ...
```

#### 4- (20 valores em 100)

Considere agora que se pretende determinar o impacto desta arma num alvo, determinando o alcance da arma de acordo com o procedimento **alcance-projetil** realizado na alínea 2.

Desenvolva o procedimento **atira-alvo** que deverá mostrar no ecrã o texto da seguinte tabela:

*Obs.: Considere o procedimento **alcance-projetil** mesmo que não o tenha feito.*

Alcance	Texto / Mensagem
<= 100 m	"Acertou"
<= 500 m	"Danificou"
<= 1000 m	"Perto"
> 1000 m	"Longe"

Observe os seguintes exemplos de utilização deste procedimento:

```
> (atira-alvo 100 30 800)
Acertou
> (atira-alvo 100 30 600)
Danificou
> (atira-alvo 100 30 2000)
Longe
```

Completar o procedimento **atira-alvo**:

```
(define atira-alvo
  (lambda (v0 angulo dist-alvo)
    ...
```

### Trabalhar com recursividade...

#### 5- (20 valores em 100)

Desenvolveu-se um novo tipo de projétil que salta um determinado número de vezes. Cada vez que salta fica apenas com 80% da velocidade inicial no salto anterior, mantendo-se o ângulo de elevação.

Pretende-se desenvolver o procedimento **arma-saltitante** que determina o alcance deste novo projétil. Este procedimento possui três parâmetros: a velocidade inicial **v0**, o **ângulo** de elevação e o **numero** de vezes que o projétil salta. Este procedimento devolve o número correspondente ao novo alcance do projétil.

*Obs.: Considere o procedimento **alcance-projetil** mesmo que não o tenha feito.*

Observe os seguintes exemplos de utilização deste procedimento:

```
> (arma-saltitante 100 30 1)
883.699391616774
> (arma-saltitante 100 30 2)
1449.2670022515094
> (arma-saltitante 100 30 3)
1811.23027305774
> (arma-saltitante 100 30 0)
0
```

Completar o procedimento **arma-saltitante**:

```
(define arma-saltitante
  (lambda (v0 angulo numero)
    ...
```

#### 6- (15 valores em 100)

Pretende-se agora desenvolver um código secreto para as comunicações de controlo do canhão.

Os números são codificados em conjuntos de dois dígitos, que multiplicados dão as parcelas que formam o número, de acordo como seguinte exemplo:

**1234567** é convertido para **7\*6 + 5\*4 + 3\*2 + 1\*0**

O procedimento **codigo** recebe apenas um parâmetro, o número a descodificar, e retorna o número descodificado.

Observe os seguintes exemplos de utilização deste procedimento:

```
> (codigo 123456)
44
> (codigo 123)
6
> (codigo 102030)
0
```

Completar o procedimento **codigo**:

```
(define codigo
  (lambda (numero)
    ...
```