



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Licenciatura em Engenharia Informática e Computação
Programação I - 2004/05

Turma 2
1ª Prova Prática
Duração: 90 min.

NOTAS IMPORTANTES:

- 1 - Deve respeitar rigorosamente os nomes dos procedimentos que são indicados bem como os formatos de saída dos resultados, nomeadamente, os espaços entre os caracteres.
- 2 - Não use nunca caracteres acentuados, nem nos nomes dos procedimentos e dos parâmetros, nem nos textos visualizados.
- 3 - O código desenvolvido durante a realização da prova, contido num único ficheiro com a extensão ".scm", deve ser submetido no Moodle usando o "link" correspondente à turma em que a prova foi realizada e não o da turma a que o aluno pertence. A não observação desta regra levará a que o código submetido seja considerado errado, na totalidade.

Numa certa disciplina, a nota final de cada aluno é obtida aplicando a fórmula $\text{nota} = 0.2 \cdot t1 + 0.3 \cdot t2 + 0.5 \cdot \text{ex}$, onde **t1**, **t2** e **ex** representam, respectivamente, as notas obtidas pelo aluno em 2 trabalhos práticos e no exame final (valores na gama [0..20]).

1-
O procedimento **visu-nota-quant** tem um parâmetro inteiro, **nota**, e visualiza a nota como se ilustra a seguir.

```
> (visu-nota-quant 10)
nota = 10
> (visu-nota-quant 15.3)
nota = 15.3
> (visu-nota-quant 19.1)
nota = 19.1
>
```

Completar o procedimento **visu-nota-quant**:

```
(define visu-nota-quant
  (lambda (nota)
    ...
```

2-
O procedimento **calc-nota** tem 3 parâmetros, inteiros, **t1**, **t2** e **ex**, com o significado anteriormente indicado, e calcula a nota do aluno, usando a fórmula já referida.

```
> (calc-nota 19 15 18)
17.3
> (calc-nota 12 10 15)
12.9
> (calc-nota 12 7 8)
8.5
> (calc-nota 12 12 10)
11.0
>
```

Completar o procedimento **calc-nota**:

```
(define calc-nota
  (lambda (t1 t2 ex)
    ...
```

3-
A nota final a atribuir é a que se obtém usando o procedimento **calc-nota** arredondada para o inteiro mais próximo. Em Scheme, existe um procedimento primitivo (**round x**) que devolve o inteiro mais próximo de **x**, mas no caso de existirem 2 inteiros igualmente próximos (parte decimal de **x** igual a 0.5) devolve o que for par; por exemplo, (**round 10.5**) devolve **10** e (**round 11.5**) devolve **12**. Não é isto que se pretende neste caso; pretende-se que desenvolva um procedimento (**arredonda x**) que no caso de a parte decimal de **x** ser 0.5 arredonde sempre para o inteiro superior.

```
> (arredonda 15.3)
15.0
> (arredonda 17.6)
18.0
> (arredonda 10.5)
11.0
> (arredonda 11.5)
12.0
>
```

Completar o procedimento **arredonda**:

```
(define arredonda
  (lambda (x)
    ...
```

4-

Pretende-se também atribuir uma classificação qualitativa ao aluno. A classificação qualitativa é atribuída com base na classificação quantitativa arredondada. Apresenta-se ao lado uma tabela com a relação entre as classificações quantitativas e qualitativas.

Classificação quantitativa	Classificação qualitativa
[18..20]	muito bom
[14..17]	bom
[10..13]	suficiente
[0..9]	mau

Caso seja passada ao procedimento uma classificação fora da gama de valores [0..20] pretende-se que seja visualizada a mensagem **"erro"**, no lugar da nota qualitativa, como se ilustra a seguir.

```
> (visu-nota-qualit 19.0)
nota qualitativa = muito bom
> (visu-nota-qualit 15.0)
nota qualitativa = bom
> (visu-nota-qualit 11.0)
nota qualitativa = suficiente
> (visu-nota-qualit 8.0)
nota qualitativa = mau
> (visu-nota-qualit 21.0)
nota qualitativa = erro
>
```

Completar o procedimento **visu-nota-qualit**:

```
(define visu-nota-qualit
  (lambda (nota-arredond)
    ...
```

5-

O procedimento **classificacao** tem 3 parâmetros, as classificações **t1**, **t2** e **ex**, já referidas, e usando os procedimentos anteriormente desenvolvidos, deve:

- calcular a nota quantitativa;
- arredondá-la para o inteiro mais próximo, usando o procedimento (**arredonda x**);
- visualizar a nota quantitativa arredondada;
- visualizar a nota qualitativa correspondente à nota quantitativa arredondada.

```
> (classificacao 19 17 18)
nota = 18.0
nota qualitativa = muito bom
> (classificacao 15 17 16)
nota = 16.0
nota qualitativa = bom
> (classificacao 10 8 7)
nota = 8.0
nota qualitativa = mau
> (classificacao 10 10 11)
nota = 11.0
nota qualitativa = suficiente
> (classificacao 12 10 100)
nota = 55.0
nota qualitativa = erro
>
```

Completar o procedimento **classificacao**:

```
(define classificacao
  (lambda (t1 t2 ex)
    ...
```