



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

Auto Teste  
AT01  
**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 use nunca caracteres acentuados, nem nos nomes dos procedimentos nem dos parâmetros.
- 3 - Utilize comentários só "with Semicolons" e nunca "with a Box".
- 4 - 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 prova realizada. A não observação desta regra levará a que o código submetido não possa ser avaliado.

Procurou-se um conjunto de exercícios que pudessem ser desenvolvidos sem recorrer a expressões de decisão (if ou cond) ou de criação de variáveis locais (let).

Frações

1- O procedimento **visu-fraccac** tem dois parâmetros, inteiros, **n** e **d**, considerados o numerador e denominador de uma fração e visualiza a fração como se indica. Não interessa o valor que devolve.

```
> (visu-fraccac 2 4)
frac: 2/4 ; <---- NOTE que só existe um espaço entre ":" e "2"
> (visu-fraccac 1 2)
frac: 1/2 ; <--- visu-fraccac NÃO TERMINA com newline
> (visu-fraccac 0 2)
frac: 0/2
> (visu-fraccac 2 0)
frac: 2/0
> (visu-fraccac 4 4)
frac: 4/4
```

Completar o procedimento **visu-fraccac**:

```
(define visu-fraccac ; <----- NOTE como está escrita a palavra fraccac...
  (lambda (n d)
    ...
```

2- O procedimento **reduz** tem dois parâmetros, inteiros, **n** e **d**, considerados o numerador e denominador de uma fração e visualiza a fração reduzida, como se indica. Não interessa o valor que devolve.

```
> (reduz 2 4)
frac: 1/2
> (reduz 1 2)
frac: 1/2
> (reduz 0 2)
frac: 0/1
> (reduz 4 4)
frac: 1/1
> (reduz 8 2)
frac: 4/1
```

Pista: Determinar o máximo divisor comum do numerador e do denominador (em Scheme existe **gcd** para calcular o máximo divisor comum) e dividir cada um deles pelo divisor determinado.

Completar o procedimento **reduz**, sabendo que este procedimento usa o procedimento **visu-fraccac**:

```
(define reduz
  (lambda (n d)
    ...
```

3- O procedimento **multiplicar-fraccoes** tem quatro parâmetros, inteiros, **n1**, **d1**, **n2** e **d2**, considerados o numerador e denominador de duas frações e visualiza o produto das duas frações no formato gerado pelo procedimento **reduz**, como se indica. Não interessa o valor que devolve.

```
> (multiplicar-fraccoes 1 4 2 4)
frac: 1/8
> (multiplicar-fraccoes 3 2 2 1)
frac: 3/1
> (multiplicar-fraccoes 1 4 3 4)
frac: 3/16
> (multiplicar-fraccoes 1 0 3 4)
frac: 1/0
> (multiplicar-fraccoes 1 4 3 0)
frac: 1/0
> (multiplicar-fraccoes 1 4 0 4)
frac: 0/1
> (multiplicar-fraccoes 0 4 0 6)
frac: 0/1
>
```

Pista: A fração correspondente ao produto de duas frações, ainda não reduzida, tem como numerador  $n1 * n2$  e como denominador  $d1 * d2$ .  
Situações do tipo (multiplicar-fraccoes 0 4 6 0) não ocorrerão.

Completar o procedimento **multiplicar-fraccoes**, sabendo que este usa o procedimento **reduz**:

```
(define multiplicar-fraccoes      ; <----- NOTE como está escrita a palavra fraccoes...
  (lambda (n1 d1 n2 d2)
    ...
```

4- O procedimento **dividir-fraccoes** tem quatro parâmetros, inteiros, **n1**, **d1**, **n2** e **d2**, considerados o numerador e denominador de duas frações e visualiza o resultado da divisão das duas frações num formato equivalente ao gerado pelo procedimento **reduz**, como se indica. Não interessa o valor que devolve.

```
> (dividir-fraccoes 2 4 1 4)
frac: 2/1
> (dividir-fraccoes 1 4 3 4)
frac: 1/3
> (dividir-fraccoes 1 0 3 4)
frac: 1/0
> (dividir-fraccoes 1 4 3 0)
frac: 0/1
> (dividir-fraccoes 1 4 0 4)
frac: 1/0
> (dividir-fraccoes 0 4 6 0)
frac: 0/1
>
```

Pista: dividir uma fração por outra torna-se fácil recorrendo ao procedimento **multiplicar-fraccoes**.  
Situações do tipo (dividir-fraccoes 0 4 0 6) não ocorrerão.

Completar o procedimento **dividir-fraccoes**, sabendo que usa o procedimento **multiplicar-fraccoes**:

```
(define dividir-fraccoes      ; <----- NOTE como está escrita a palavra fraccoes...
  (lambda (n1 d1 n2 d2)
    ...
```

----- FIM da Prova Prática -----