

# Projeto de funções - Prática

Marco A L Barbosa

malbarbo.pro.br

- 1) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
;; InteiroPositivo -> Boolean  
;;  
;; Produz #t se uma pessoa com idade id é isento da  
;; tarifa de transporte público, isto é, tem menos  
;; que 18 anos ou 65 ou mais. Produz #f caso contrário.
```

(examples

```
(check-equal? (isento-tarifa? 17) #t)  
(check-equal? (isento-tarifa? 18) #t)  
(check-equal? (isento-tarifa? 50) #f)  
(check-equal? (isento-tarifa? 65) #t)  
(check-equal? (isento-tarifa? 70) #t))
```

```
(define (isento-tarifa? id)  
  #f)
```

- 2) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
;; Inteiro -> Inteiro  
;;  
;; Conta a quantidade de dígitos de n.  
;; Se n é 0, então devolve zero.  
;; Se n é menor que zero, então devolve a quantidade  
;; de dígitos do valor absoluto de n.
```

(examples

```
(check-equal? (quantidade-digitos 121) 3)  
(check-equal? (quantidade-digitos 0) 1)  
(check-equal? (quantidade-digitos -1519) 4))
```

```
(define (quantidade-digitos n)  
  0)
```

- 3) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
;; InteiroPositivo -> Boolean  
;;  
;; Produz #t se uma pessoa com a idade n é supercentenária,  
;; isto é, tem 110 anos ou mais, #f caso contrário.
```

(examples

```
(check-equal? (supercentenario 101) #f)  
(check-equal? (supercentenario 110) #f)  
(check-equal? (supercentenario 112) #t))
```

```
(define (supercentenario n)  
  #f)
```

- 4) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
;; String -> String  
;;  
;; Transforma a data de entrada do formato "dia/mes/ano"  
;; para o formato "ano/mes/dia".  
;;  
;; Requer que o dia e o mês tenham dois dígitos e que  
;; o ano tenha quatro dígitos.
```

(examples

```
(check-equal? (data-dma->data-amd "19/07/2023") "2023/07/19")  
(check-equal? (data-dma->data-amd "01/01/1980") "1980/01/01")  
(check-equal? (data-dma->data-amd "02/02/2002") "2002/02/20"))
```

```
(define (data-dma->data-amd data)  
  data)
```

- 5) Escreva a especificação para a seguinte implementação de função. Observe que a especificação sozinha deve ser suficiente para um desenvolvedor fazer uma implementação.

```
(define (aumenta valor porcentagem)  
  (* valor (+ 1 (/ porcentagem 100))))
```

- 6) Escreva a especificação para a seguinte implementação de função. Observe que a especificação sozinha deve ser suficiente para um programador fazer uma implementação.

```
(define (tamanho-nome nome)  
  (cond  
    [(<= (string-length nome) 4) "curto"]  
    [(<= (string-length nome) 10) "médio"]  
    [else "longo"])))
```

- 7) Projete uma função que adicione um ponto final a uma frase se ela não acabar com um.
- 8) Projete uma função que determine se uma palavra tem um traço ("-") no meio, como por exemplo, "lero-lero". Não use `if` nem `cond`.
- 9) Projete uma função que encontre o máximo entre três números dados. Não use a função pré-definida `max`.
- 10) Projete uma função que receba como parâmetro uma string e um número natural  $n$  e substitua os primeiros  $n$  caracteres da string por  $n$  letras  $x$ .
- 11) Faça a execução passo a passo das funções definidas nos exercícios anteriores para alguns exemplos. Para isto você deve criar um novo arquivo, mudar a linguagem para "Beginning Student" e copiar o código da função e da chamada de exemplo.