

Fundamentos - Problemas

Marco A L Barbosa

malbarbo.pro.br

Para implementar os exercícios a seguir, use apenas as funções primitivas presentes no material “Resumo da linguagem Racket” disponível na página da disciplina.

- 1) Faça uma função chamada `area-retangulo` que recebe dois argumentos, a `largura` e a `altura` de um retângulo, e calcula a sua área. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (area-retangulo 3 5)
15
> (area-retangulo 2.0 2.5)
5.0
```

- 2) Faça uma função chamada `produto-anterior-posterior` que recebe um número inteiro `n` e calcula o produto de `n`, `n + 1` e `n - 1`. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (produto-anterior-posterior 3)
24
> (produto-anterior-posterior 1)
0
> (produto-anterior-posterior -2)
-6
```

- 3) Faça uma função chamada `so-primeira-maiuscula` que recebe uma palavra não vazia (string) como parâmetro e crie uma nova string convertendo a primeira letra da palavra para maiúscula e o restante da palavra para minúscula. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (so-primeira-maiuscula "paula")
"Paula"
> (so-primeira-maiuscula "ALFREDO")
"Alfredo"
```

- 4) Faça uma função chamada `exclamacao` que recebe dois argumentos, uma string `frase` e um número natural `n`, e produz a mesma frase adicionando `n` pontos de exclamação no final da frase. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (exclamacao "Nossa" 3)
"Nossa!!!"
> (exclamacao "Que legal" 1)
"Que legal!"
> (exclamacao "Nada" 0)
"Nada"
```

- 5) Faça uma função chamada `par?` que recebe um número natural `n` e indica se `n` é par. Um número é par se o resto da divisão dele por 2 é igual a zero. Não use `if` nem `cond` e nem a função pré-definida `even?`. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (par? 3)
#f
```

```
> (par? 6)
#t
```

- 6) Faça uma função chamada `tres-digitos?` que recebe um número natural `n` e verifica se `n` tem exatamente 3 dígitos. Não use `if` nem `cond`. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (tres-digitos? 99)
#f
> (tres-digitos? 100)
#t
> (tres-digitos? 999)
#t
> (tres-digitos? 1000)
#f
```

- 7) Faça uma função `maximo` que encontre o máximo entre dois números dados. Não use a função pré-definida `max`. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (maximo 3 5)
5
> (maximo 8 4)
8
> (maximo 6 6)
6
```

- 8) Faça uma função chamada `ordem` que recebe três números distintos, `a`, `b` e `c` e determina se a sequência `a`, `b`, `c` está em ordem crescente, decrescente ou não está em ordem. Use os operadores relacionais com três argumentos. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
> (ordem 3 8 12)
" crescente "
> (ordem 3 1 4)
" sem ordem "
> (ordem 3 1 0)
" decrescente "
```