Paradigmas de Linguagem de Programação

## Exercícios Paradigma Funcional - Scheme

**Observações:**

* O trabalho pode ser feito em duplas ou com o grupo de trabalho da disciplina, com consulta ao material de aula, apostilas e Internet;
* Deve ser realizado em Linguagem Lisp, com o dialeto Scheme;
* Utilize o interpretador Scheme Online, disponível em: <https://repl.it/languages/scheme>
* Deverá ser entregue até o dia de hoje, 12 de outubro de 2016, até as 23:00 pelo Moodle;
* Organize seu tempo e Bom Trabalho!!!

1. (1.0 Ponto) Considere a seguinte interpretação em Scheme:

# (define (f1 x) (\* x x)) (f1 5)

**25**

**(define f2 f1)**

* 1. Qual o valor de (f2 5)? Justifique sua resposta.

**25, a função “f1” faz a multiplicação do parâmetro passado por ele mesmo. A função “f2” assume a função “f1”. Então quando executado (f2 5), “f2” assume o parâmetro 5, como ela chama a função “f1”, logo a função “f1” assume, no caso “5”, realizando a multiplicação \* 5 5, que resulta em 25.**

* 1. Suponha que agora defina o procedimento f2, da seguinte maneira:

# (define (f2 x) (+ x 10))

Quais os valores de (f1 5) e de (f2 5)? Justifique sua resposta.

**(f1 5) não utiliza a função “f2”, executando somente a primeira função “f”, que realiza a multiplicação do parâmetro passado, por ele mesmo, logo foi passado “5”, então \* 5 5 = 25;**

**(f2 5) nessa alteração a função “f2” realiza somente a soma do parâmetro passado + 10, logo foi passado “5”. Então + 5 10 = 15.**

1. (1.5 Pontos) O supermercado Gaúcho Legal precisa de um programa que calcule o valor de um saco de moedas. Sabe-se que é possível encontrar moedas de: 05, 10, 25 e 50 centavos e também moedas de 1 real. Apresente a solução computacional para essa contagem (apresente o teste de mesa).

**(define (soma\_aux lista sm)**

**(if (null? lista)**

**sm**

**(soma\_aux (cdr lista)**

**(+ sm**

**(if (< (car lista) 2) (\* (car lista) 100)**

**(car lista))))))**

**(define (soma lista) (soma\_aux lista 0))**

**(/(soma (list 1 10 25))100)**

**//1**

(2.0 Pontos) Implemente uma função que receba duas listas como parâmetro e retorne “Verdadeiro” se elas forem iguais e “Falso”, caso contrário. Apresente o teste de mesa.

**(define (compara lista1 lista2)**

**(if (equal? lista1 lista2)**

**"Verdadeiro" "Falso") )**

**(compara (list 1 1 5 2 3 ) (list 1 1 5 2 3))**

**// a funcao assume os dois parametros passados, que são listas, após receber ela realiza um teste lógico com as duas listas recebidas, através da função equal? Que compara se os dados do primeiro átomo e do Segundo são iguais, em caso de resultado positivo do teste lógico executa o então do teste lógico if, que é a string “verdadeiro”, caso seja falso o resultado executa o “Else” do teste lógico que é a string “falso”.**

1. (2.0 Pontos) Implemente uma função em Scheme, que recebe uma lista e um elemento qualquer e verifica se o elemento pertence à lista ex.: (Pertence 1 (list 3,7,4,2)) => False (resposta).

**(define (compara x lista1)**

**(if (null? lista1)**

**"=> False nao pertence"(if (eq? x (car lista1))**

**"=> True pertence"(compara x (cdr lista1)))))**

**(compara 5 (list 1 2 3))**

1. (2.0 Pontos) Implemente uma função em Scheme, que recebe um número inteiro positivo n, e devolve a soma de todos os números até n.

**(define (soma n)**

**(if (zero? n)**

**0**

**(+ n (soma (- n 1)))))**

1. (1.5 Pontos) Implemente uma função em Scheme que dada uma lista como parâmetro devolva a quantidade de elementos que essa possui.

**(define (zera lista cont)**

**(if (null? lista) cont (zera (cdr lista) (+ cont 1))))**

**(define (soma lista) (zera lista 0))**

**(soma (list 20 50 3 5 80 120))**

Profª. Maria Adelina Raupp Sganzerla – [masganzerla@gmail.com](mailto:masganzerla@gmail.com)