

**UNIVERSIDADE FEDERAL FLUMINENSE
PÓLO UNIVERSITÁRIO DE RIO DAS OSTRAS
INSTITUTO DE CIÊNCIA E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

V.S. de Linguagens de Programação – 1º. Sem de 2011

Prof.: Carlos Bazilio

Aluno:

Matrícula:

1. (2.0 ptos) A linguagem Haskell é uma linguagem funcional similar, semanticamente, à linguagem Scheme. Sintaticamente, esta possui características que a aproximam da matemática, com casamento de padrões similar à Prolog (uma linguagem de programação baseada em lógica). Supondo a questão de manipulação de conjuntos abordada na P2 deste semestre, observe as versões em Scheme e Haskell das implementações da função `pertence` de conjuntos.

| Scheme | Haskell |
|---|---|
| <pre>(define pertence? (lambda (el conj) (if (null? conj) '#f (if (= (car conj) el) '#t (pertence? el (cdr conj))))))</pre> | <pre>pertence el [] = False pertence el (cab:resto) el == cab = True otherwise = pertence el resto</pre> |

Baseado neste exemplo, crie em Haskell:

- a) Uma função que retorne o tamanho de um conjunto, lembrando que não há repetição entre os elementos num conjunto; ou seja, o tamanho do conjunto $\{1, 2, 3, 1\}$ é 3.
 - b) Uma função que implemente a união entre conjuntos.
2. (3.0 ptos) Considere o código abaixo que exemplifica o uso de um sistema de vendas:

```
public class SistemaVendas {  
    public static void main(String[] args) {  
        Estoque produtos = new Estoque();  
        produtos.adicionaProduto(new Produto ("monitor", 500));  
        produtos.adicionaProduto(new Produto ("telefone", 150));  
        produtos.adicionaProduto(new Produto ("teclado", 70));  
        produtos.adicionaProduto(new Produto ("mouse", 50));  
  
        CarrinhoCompra carrinho = new CarrinhoCompra(produtos);  
  
        carrinho.adicionaItem("monitor", 2);  
        carrinho.adicionaItem("telefone", 5);  
        carrinho.adicionaItem("teclado", 2);  
  
        System.out.println("A soma dos produtos é: " +  
            carrinho.calculaTotal());  
    }  
}
```

Apresente uma possível definição para as classes Estoque, Produto (cada produto possui um nome e seu valor) e CarrinhoCompra, as quais compõe este sistema de vendas. Observe que, na adição de um item no carrinho de compras, além do nome do produto, é fornecida a quantidade de itens a serem adquiridos. O método *calculaTotal()* retorna a soma total de produtos existentes no carrinho.

3. (3.0 pts) Suponha o código em C a seguir que manipula uma lista genérica :

```

#include "stdio.h"
#include "stdlib.h"

typedef struct lista_gen {
    void * info;
    void (* impressao)(void *); // Campo da estrutura que representa uma função
    struct lista_gen* prox;
} ListaGen;

void insere(ListaGen **l, void* p, void impr(void *)) {
    ListaGen* n = (ListaGen *) malloc(sizeof(ListaGen));
    n->info = p;
    n->impressao = impr;
    n->prox = *l;
    *l = n;
}

void imprimeString (void *p) {
    printf("%s\n", (char *)p);
}

void imprimeNumero (void *p) {
    printf("%d\n", *((int *)p));
}

void exibe(ListaGen *l) {
    ListaGen *laux;
    for (laux = l; laux != NULL; laux=laux->prox) {
        laux->impressao(laux->info);
    }
}

main() {
    ListaGen* l = NULL;
    int *x = (int *)malloc(sizeof(int));
    (*x) = 10;
    insere(&l, x, imprimeNumero);
    char *nome = "bazilio";
    insere(&l, nome, imprimeString);
    exibe(l);
}

```

- a) Qual é o resultado da execução deste programa?
 - b) Reescreva este programa, em Java por exemplo, usando conceitos de OO sempre que possível.
 - c) Suponha que queiramos adicionar à lista valores que representam pontos em 3D (x, y, z). Altere este programa de maneira isto seja possível.
4. (2.0 pts) Fale, de maneira organizada e clara, tudo que sabe a respeito de alocação dinâmica: i) situações onde ocorre; ii) razões e vantagens; iii) desvantagens e mecanismos amenizam ou contornam esta situação.