Universidade Estadual de Maringá – Departamento de Informática
11917/31 – Programação Funcional – Informática
Prof. Marco Aurélio

Trabalho 03

1) Projete uma **função não recursiva** que encontre todos os nomes de uma lista de nomes que tenham mais que cinco caracteres. Use a função `string.length` para determinar o número de caracteres de uma string.

2) Projete uma **função não recursiva** que receba como parâmetro uma lista de strings e um índice `i` e devolva uma lista com o caractere na posição `i` (ou vazio se o índice for inválido) de cada string da lista. Use a função `string.slice` para determinar o caractere no índice `i` de uma string.

```
> multi_get(["casa", "mouse", "da"], 3)
["a", "s", ""]
```

3) Projete uma **função não recursiva** que crie uma nova lista removendo as primeiras ocorrências dos elementos repetidos de uma lista de entrada. Use a função `list.contains` parra verificar se um elemento está em uma lista.

```
> unicos([7, 1, 4, 7, 1])
[4, 7, 1]
```

4) Projete uma **função recursiva em cauda** que conte quantas vezes um elemento aparece em uma lista.

Referência

```
/// Devolve a quantidade de caracteres de *s*.
fn string.length(s: String) -> Int

fn string_length_examples() {
  check.eq(string.length(""), 0)
  check.eq(string.length("casa"), 4)
}

/// Devolve uma substring de *s* começando em *inicio* e pegando os próximos
/// *tam* caracteres ou até o fim de *s*, o que vier primeiro.
/// Se *inicio* é negativo, começa a partir do fim de *s*.
/// Se *tam* é negativo, devolve "".
fn string.slice(s: String, inicio: Int, tam: Int) -> Int

fn string_slice_examples() {
  check.eq(string.slice("funcional", 2, 3), "nci")
  check.eq(string.slice("funcional", 2, -1), "")
  check.eq(string.slice("funcional", 3, 50), "cional")
  check.eq(string.slice("funcional", -3, 2), "na")
}

/// Devolve True se *v* está em lst, False caso contrário.
fn list.contains(lst: List(a), v: a) -> Bool

fn list_contais_examples() {
  check.eq(list.contains([3, 1, 7], 1), True)
  check.eq(list.contains([3, 1, 7], 0), False)
}
```