2024-09-27

5118014-02 Programming Language Theory :: Test 1

Problem 1 (20 points). Define a function **`count`** using **`filter`** and **`fold`**, that takes a list of integers and an integer ***e*** as input and returns the number of occurrences of ***e*** in the list. Ensure that the implementation of **`count`** is simple as possible. For example:

```
assert(count(List(2,3,2,1,2,4,5,1), 2) == 3)
```

```
def filter (l: List[Int], f: Int => Boolean) : List[Int] = { ··· }
def fold (l: List[Int], n: Int, f: (Int, Int) => Int) : Int = { ··· }
def count (l: List[Int], e : Int) : Int = {




}
```

Problem 2 (30 points). Define a function **`map`** for a list of integer lists (**`map`** must not use the built-in *map* function of the Scala standard library). **`map`** must take a list of integer lists (i.e., `List[List[Int]]`) and a function **`f : Int => Int`** as input, and generate a list of integer lists by applying **`f`** to each integer element. For example:

```
assert(map(List(List(1,2),List(3)), _ + 1) == List(List(2,3),List(4)))
```

Name:_____ Student number: _____

Problem 3 (30 points). Write a function **`polynomial`** that takes a list of integers **`List`**$(a_0, a_1, ..., a_n)$ as input and generates a function that evaluates $a_0x^{n-1} + a_1x^{n-2} + \cdots a_{n-1}x + a_n$ for a given integer x For example:

```
assert(polynomial(List(3,2,1)(2) == 17)
```

Problem 4 (20 points) Explain what tail recursion is.

Name:_____ Student number: _____