2024-09-26

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 (40 points). Implement a function **`append`** that concatenates two given lists of integers in order. For example:

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

Name:_____ Student number: _____

Problem 3 (20 points). Write a function **`listgen`** that takes an integer v as input and generates a function as output; the generated function takes an integer n as input and produces a list that consists of n occurrences of v. For example:

```
assert(listgen(5)(3) == List(5, 5, 5))
```

Problem 4 (20 points) Explain what Algebraic Data Type (ADT) is.

Name:_____ Student number: _____