EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mathematisch-Naturwissenschaftliche Fakultät

Programmiersprachen

Prof. Klaus Ostermann

Steven Lolong
steven.lolong@uni-tuebingen.de

Object Meets Function

Assignment 3 – Winter Semester 23/24

Tübingen, 13. Dezember 2023

Handin Please submit this homework until Thursday, December 21st 2023 via email to Steven Lolong (steven.lolong@uni-tuebingen.de) before 24:00.

Email Format Use this format for the email's subject and file's name: **OmF-W23/24-Assign**[no]**-**[YourName]

Task 1: For-loop and HoF (8 points)

1.1 Add and Substract (4 points)

There are two supporting functions for you to complete this task.

```
// successor
def succ(i : Int) : Int = i + 1

// predecessor
def pred(i : Int) : Int = i - 1
```

There are two HoFs below that you should complete. These functions will do the basic math operation for addition and subtraction. However, you can't use the operator + for the add function and - for the subtract function. Otherwise, you can use two support functions.

```
  def add(x: Int, y: Int, helper: Int => Int) : Int = { ... }
  def subtract(x: Int, y: Int, helper: Int => Int) : Int = { ... }
```

1.2 Multiplication (4 points)

Please complete the incomplete function below for the multiplication of two numbers, but you can't use the * operator. Otherwise, you can use add function from task 1.1. But, in this task, you must concern about the negative number.

```
def mult(x: Int) (y: Int) (helper: (Int, Int, Int => Int) => Int) = { ... }
```

*note:

1. Remember to make an example function application for both task 1.1 and task 1.2.

Task 2: Recursion (10 Points)

2.1 Add and Subtract (6 Points)

Some functional programming doesn't support for-looping but uses the recursion method for replacement of for-looping. Assume Scala is one of them. Please write functions for task 1.1 (add and subtract), but you can't use the + and - operator, and avoid using ”for” and ”while”. What you need is only helper ”succ” and ”pred” function.

2.2 Vowels in a string (4 Points)

Write a recursive function that returns the number of vowels in a string. Use the following quote, by Dalai Lama, for your testing:

”The purpose of our lives is to be happy.”

*vowels = A, E, I, O, U, a, e, i, o, u

Task 3: More about recursion (6 Points)

Write a recursive function that generates the following pattern of letters. Make sure that the user enters an integer between 1 and 26. For example, if the inputted number is 6, then the pattern generated is:

```
ABCDEF
ABCDE
ABCD
ABC
AB
A
A
AB
ABC
ABCD
ABCDE
ABCDEF
```

Bonus (2 Points)

Please write function(s) to do task 3 but instead of using a recursive function, please solve it using for-loop!