CS_205 Declarative Programming 26 October 2017
Monika Seisenberger

Assignment 1: Due 9 November 2017, 11am

1. This coursework will be submitted in small groups of two or three. As a group, you are asked to get together, discuss your ideas, plan the solutions, compare your solutions, etc. Planned is that each group presents their work after submission to the lecturer. Everyone needs to be able to explain the main ideas in the submission.
2. Please register in pairs of two or three on blackboard by end of October, the link will be open on Friday 27th October.
3. Marks will be awarded for both correct functionality and good code quality as well as for complying with the instructions.
4. Please submit exactly 1 Haskell file and add the answers to additional questions as comments.
5. University takes plagiarism very seriously. With the submission you take responsibility that this is the group's own solution, and that the group has implemented the code, rather than somebody having taken it from somewhere else. After submission you have until the end of the day to notify me if you do not want to take ownership of certain parts in your submission.
6. Your names and student numbers as well as a sentence that this the group's own work must be written at the beginning of the file.

Question 1. a) Write two different functions `max1` and `max2` that compute the maximum of three numbers (without using built-in functions for computing the maximum).

(Note: your two functions should implement different algorithms.)

Run your programs with your student numbers (if you are only two in your group use 876543 as a third number), and include the result as comment into your solution.

b) The following demonstrates a method how you could check the correctness of your programs: Add the following code to the top of your file `import Test.QuickCheck` and write a function as below (please complete its type) that compares the two implementations of your howManyAboveAverage functions:

```
checkcorrectness x y z =
     max1 x y z  == max2 x y z
```

In the command line then do the check by typing.

```
quickCheck checkcorrectness
```

Include the response of Haskell as a comment.

Also think about possible weaknesses of this correctness; how could it be improved?

[10 marks]

Question 2. Define a function that tests whether a given list is sorted. Then take three different sorting algorithms (insertionSort from the lab, quicksort and mergesort from the coursenotes). Write one function that takes as input a list of number and checks whether all three sorting algorithms yield a sorted list. It should be easy to add another sorting algorithm. [Ideally you list all sorting algorithms in a list which you then can extend once another algorithm is given.] Add as a comment how you call your function and what its output is.

[8 marks]

Question 3. a) Using the functions `factors` and `prime` in chapter 5 of the course, define a function `primesbelow` that produces all prime numbers that are below a given input n.

b) Given

```
allprimes :: [Integer]
allprimes = [x | x<- [2..], prime x]
```

define an infinite list `primeTest:: [Bool]` such that the nth position of the list is `True` if n is prime, and `False` otherwise. Then, produce an infinite list such that the nth position is the pair (n, b) where the value b is `True` if n is prime and `False` otherwise. Hint: use `map` and `zip`.

c) Finally, add a function that, given n, computes how many prime twins are amongst the first n prime numbers.

d) Use this to answer the question: how many prim twins are there amongst the first 2000 prime numbers? Add your result as a comment.

[8 marks]

Question 4. Suppose you are at your home town which is the city numbered by your student number. Your goal is to reach city 1. The traveling instructions are as follows: If you are at city k and k is an even number, then go to city $\frac{k}{2}$. If k is odd, then go to city $3k + 1$. Compute the list of all cities you visit on your journey. How many cities do you visit and which is the city with the largest number on your journey? Add the list and the answers to these two questions as a comment for each student number.

[8 marks]

Question 5. (Phonetic Search) The following is a programming test for job applicants provided by leading UK company: Your program must read a list of surnames (surname.txt) from standard input, one per line. The command-line argument to your program will be also a list of surnames. For each of these surnames you must print out all of the names from surnames.txt that match those from the command-line. If the command-line arguments were ”Smith” and ”Jones”, and each matched three names in the input data, then you should print out (with the appropriate punctuation as shown):

Smith: Smidt, Smith, Smyth

Jones: Johns, Jonas, Jones

The matching algorithm is as follows:

1. All non-alphabetic characters are ignored
2. Word case is not significant
3. After the first letter, any of the following letters are discarded: A, E, I, H, O, U, W, Y.
4. The following sets of letters are considered equivalent
   - A, E, I, O, U
   - C, G, J, K, Q, S, X, Y, Z
   - B, F, P, V, W
   - D, T
   - M, N
   - All others have no equivalent
5. Any consecutive occurrences of equivalent letters (after discarding letters in step 3) are considered as a single occurrence

The rules should be applied in that order. So, given a file ”surnames.txt” that contains the following data:

```
Smith
Smyth
Smythe
Smid
Schmidt
Smithers
Jonas
Johns
Johnson
Macdonald
Nest O'Malett
Ericsson
Erikson
Saunas
Van Damme
```

Running your program from the command-line like this:

```
MyProgram [Jones, Winton]
```

Should print out:

```
Jones: Jonas, Johns, Saunas
...
```

Hint: As a start, or for a solution with almost full marks, I recommend to provide the names in surname.txt as a list within your program. The file input is something that will require some research. The main work of this question should consist of implementing the matching algorithm correctly.

[10 marks]

Overall: Programming Style, Presentation of work, Adherance to rules.

[6 marks]

Note that functionally, completeness, and elegance is marked in each question.