Eastern Mediterranean University

CMSE318-CMPE410 Principles of Programming Languages
Spring 2023-2024

Assignment 5

Functional Programming with Haskell

To be done in groups of two. Pick your partner!

First, install the Haskell interpreter provided in the resources. Documentation will automatically be installed as well. Review the lecture notes on Haskell before you attempt this assignment. You may also find this site useful: http://learnyouahaskell.com/chapters.

Using an appropriate text editor (e.g. Notepad++ available at https://notepad-plus-plus.org/downloads/), create a file with .hs extension that contains Haskell definitions for the following functions.

1. **div_by_3** ***some_list***: returns the list of numbers that are divisible by 3. For example, **div_by_3** [3, 5, 4 ,2, 8, 15] should return the list [3,15]. **Constraint:** Use a ***list comprehension*** in the implementation of this function.

2. **div_by_2** ***some_list***: returns the list of numbers that are divisible by 2. For example, **div_by_2** [3, 5, 4 ,2, 8, 15] should return the list [4,2,8]. **Constraint:** Implement this function ***recursively***.

3. **count** ***elem a_list***: returns how many *elem*s are in *a_list*. For example, **count** 'b' ['a', 'b', 'c', 'b', 's'] should return 2.

4. **triple** ***list***: Return a list where all values in the list are triple the values in the original list. For example, **triple** [2,4,5,1] should return [6,12,15,3]. **Constraint:** use the library function **map**, and define a helper function that just multiplies its parameter by 3.

5. **sort** ***n1 n2 n3***: returns a list containing the three numbers n1, n2 and n3 such that the list is sorted. E.g. **sort** 3 4 1 should return [1,3,4]. **Constraint:** Define a guarded, non-recursive function to do the job!

6. **from_to** ***n m list***: returns a slice of the list from *position n* to *position m*. E.g. **from_to** 3 5 [4,3,6,5,8,7,2,0] should return [6,5,8]. Note that we start counting at 1, not 0. **Constraint:** Study the documentation and learn about library functions **take** and **drop**. Use these functions in your definition.

7. **nth_element *n list***: returns the n^{th} element of list. Counting starts at 1. For example, **nth_element** 3 [2,5,4,6,8] should return 4. **Constraint:** Define this function recursively.

8. **mth_element *m list***: returns the m^{th} element of list. Counting starts at 1. For example, **mth_element** 3 [2,5,4,6,8] should return 4. **Constraint:** Define this function using library functions **head** and **drop**.

9. **add_lists *list1 list2***: returns the sum of the elements in the same position in each list as a new list. If one list is shorter than the other, assume that the shorter list has zero's in the "missing places". For example,
    - **add_lists** [3,4,5] [10,20,30] should return [13,24,35].
    - **add_lists** [3,4,5] [10,20,30, 88, 42] should return [13,24,35,88,42].
    - **add_lists** [3,4,5] [10] should return [13,4,5]

10. **classify *n***: returns a letter grade based upon a numeric grade, according to the following schema:
    - if $n \geq 90$, return 'A'
    - if $89 \leq n \leq 70$, return 'B'
    - if $69 \leq n \leq 50$, return 'C'
    - otherwise, return 'D'.

    For example, **classify** 87 should return 'B'.

    Constraint: Use a function with guards to implement this function.

What to hand in:

Zip the following and upload to Teams before the deadline.

1. A report containing a section for each function in the assignment containing
    a. the description of what the function does (copy from this document)
    b. the function definition itself
    c. screenshots of at least two calls with different parameters, making sure that all cases are handled

2. Your source file with .hs extension

    The filename of your uploaded file should be of the format:
    studentID1_studentID2_CMSE318_CMPE410_Assignment5.zip