Lab 8 start:

These exercises should help you get started with Haskell. You may need to refer to ch. 1 and 2 in the book for examples that will help you in solving the exercises.

1.List comprehensions

a) Outfits. A notion of basic wardrobe suggests that it is possible to create a large number of outfits using only a few "basic" clothing articles. Consider the person who owns 2 pairs of pants (jeans and khaki), 4 pairs of tops (white shirt, grey turtleneck, pink polo, green hoodie) and 3 pairs of shoes (brogues, converse, sandals). Write a list comprehension expression that generate a list of tuples of all possible outfits that can be constructed with these items.

b) You will now analyze a basic wardrobe from a). Assign resulting list from part a) to “variable” outfits. Type length outfits to see how many different outfits can be constructed from these few items.

Try adding more items to each list and observe the effect on the resulting number of outfits. If you only have budget for two more articles, does it make sense to buy more items of the same type or different type? (e.g. buy two more tops or by a top and a pair of shoes)?

Which two articles will yield the maximum increase in the number of outfits?

Now consider adding another category (say, scarves) with two elements in it (paisley scarf, knitloop). Update your expression to generate a list of 4-item outfits. How many outfits are generated now? Type up your answers as comments below the code for outfit list generating expression.

c) There’s a fancy event that people can attend solo or as a couple. Given a list l of honorifics accepted in some culture, write a Haskell expression (that includes list comprehension) that contains all possible combinations of prefixes for the guest(s). Be sure to accommodate both single and couple guests. E.g. if the original honorific list is [“Mr.”, “Ms.”], then the resulting list should be [“Mr.”, “Ms.”, “Mr. and Mr.”, “Ms. and Mr.”, “Mr. and Ms.”, “Ms. and Ms”]. Order does not matter.

For testing, use l = ["Mr.", "Ms.", "Mrs.", "Dr.", "Prof.", "Rev."]

d)NATO phonetic alphabet is used to avoid errors in military communication
https://en.wikipedia.org/wiki/NATO_phonetic_alphabet

Each character is spelled as a corresponding word in the alphabet. Assume the alphabet is stored in a list l as list of tuples with ‘Character’, “Word” correspondence, and the word

is tored in word. Write a list comprehension expression that translates word into its military spelling, i.e. where “DAY” becomes [“Delta”, “Alpha”, “Yankee”]. You can assume only uppercase characters are used to specify the word. For your convenience, NATO alphabet is provided.

2. Functions

a) Implement the remainder function that takes two integers as input and returns a remainder of dividing the first one by the second one.

b) Implement isEven function that takes one integer as input and returns True if it is even, False otherwise

c)Implement the merge function that takes two sorted lists as parameters and returns the merged sorted list.

d)Write a function removeMultiple which takes a list L and a number a as arguments, and it returns a new list containing all the numbers in L that are not multiples of a.

(Hints: (1) It may be easier to create the output list with numbers in reverse order. That is fine as is; or you can use reverse afterwards. (2) You may want to use the remainder function from part a).

3.List functions
Rewrite your expressions from Part 1 as functions. As a result, you should have 3 functions:

1. outfits
2. honorifics lst (takes in a list of possible choices)
3. nato wrd (takes in a word to be translated)

Manually test your functions by calling them with several different values.