Assignment 7

Higher-order functions

In this assignment you will continue using the GitHub features we have seen so far, and practice using the standard higher-order functions from the List module.

Instead of instructions in the file itself, this time the instructions are provided in a Markdown document: https://github.com/skiadas/ProgLangAssignments/blob/master/assignment7doc.md

As before, you should NOT specify the types of your functions, but rather let the system figure it all out. This may make the resulting types appear different than what is required, you should make sure that they are indeed the same and that the only difference is because of type aliases. In some cases the system might derive a "more general type" than the expected one, and that is OK.

As in prior assignments, you should not use any functions we have not learned about unless explicitly told to. These assignments are not about learning a number of library functions, but about delving deeply into fundamental building blocks of programming. In this specific assignment, you are asked to use many functions from the List module.

One considerable deviation from previous assignments is that in this assignment you are not allowed to create any recursive functions. You must instead rely on folding and maps to process the relevant lists.

Correctness of the solutions, including the types and whether the code loads, will count for 15 points. Your code MUST load with no errors and have the correct types. Otherwise you will receive 0 for correctness even if most of your functions are correct. Start by creating stubs for all functions so that the tests are at least executed.

Another 5 points will come from style issues, your tests and use of GitHub, including Milestones and Labels, issues and closing those issues via commits. Keep creating issues for the various parts of the assignment. assigining them to milestones and tagging them with labels. **Create new milestones for each assignment**.

1. You already have created a link to the instructor's remote repository. You need to download the updated version of the instructor's repository and merge the changes into yours. The following should do that:

   - Type: `git fetch instr`
   - Type: `git merge master instr/master`. This may give you a editor window to edit a commit message. You don't need to edit it, just "save" (writeOut) and exit/close the document. (This is probably not your prefered editor. See this page[1] for how to set up your favorite editor for use in these merge situations. I would recommend setting it to SublimeText).
   - If it reports no problems, you're ready.
   - If it reports merge conflicts, you will need to resolve those first, then make a commit. this page[2] has some instructions on how to do that. Or contact me.

[1] https://help.github.com/articles/associating-text-editors-with-git/

[2] https://help.github.com/articles/resolving-a-merge-conflict-from-the-command-line/

2. Create a new milestone for this assignment.

3. Throughout your work, create and close issues via commits as described in the instructions above and in previous homeworks. Make sure to assign each issue to the correct milestone and to give them the correct labels.

4. You will find two files in this directory:

   - assignment7sub.ml This is the main submission file. It is where you will add your code for the various functions that you need to write. There are comments in that file to show you where to add your function definitions, and to tell you what your functions should do.
   - assignment7tests.ml This is a file with a small number of tests, and you should add plenty tests your own. "Tests" are arranged as lines let ... = e where e is a an expression that is meant to evaluate to a boolean indicating if the tests succeeded or not.

5. To "run" your tests, start an OCAML session in the terminal via utop, do:

```
#use "assignment7sub.ml";;
#use "assignment7tests.ml";;
```