Assignment 1

Introduction to OCAML.

You may need to refer to the Pervasives module[1] documentation occasionally. All the functions there are available in OCAML by default. You are NOT allowed to use everything that's there unless we have talked about it or unless the assignment question asks you to look it up.

In this assignment you are asked to implement 9 functions. Each function is worth one point, and the assignment has an extra point for "style", for a total of 10 points.

The assignment expects you to use GitHub and Git to keep track of your work. The first couple of steps below will get you set up.

1. First off, if you do not have a GitHub account yet, go to https://github.com/ and set one up.

2. Next you need to "fork" the repository that has the assignments. Open your browser to https://github.com/skiadas/ProgLangAssignments and click on the "Fork" icon near the top left.

3. After a while, this will have created a "repository" under your GitHub account. You should be able to see it by going to https://github.com/yourAccountHere/ProgLangAssignments where you use your account name instead of yourAccountHere.

4. Next, you will "clone" your GitHub repository to your local computer.

    - From your terminal window, navigate to the directory you want to use as your workspace for the assignments. The next steps will create a new directory in there, so you do not need to create a new directory, just go to the location you want for that directory.
    - From your project's webpage, find the textbox near the middle that says something like https://github.com/skiadas/ProgLangAssignments.git. It will have "HTTP" to its left. Click in that textbox and copy that whole text to the clipboard.
    - Back at your terminal, type: `git clone <pasteThatLinkHere>`.
    - This should have created a new directory called "ProgLangAssignments" and you will find some homework files in it.
    - You probably will want to open the whole directory in SublimeText. After you change directory to go into that directory, you can type `subl .` to do that.

5. While in this directory, follow the instructions at https://help.github.com/articles/setting-your-username-in-git/ and https://help.github.com/articles/setting-your-email-in-git/ to set up username and email. These will be used when you "make commits".

[1] http://caml.inria.fr/pub/docs/manual-ocaml/libref/Pervasives.html

6. For now we will not do much more with Git and GitHub. Follow the remaining instructions for working on your project, and when you are done run the following from the terminal:

   - `git add .` This "prepares" your changes for committing.
   - `git status` This should show you two files ready for commit.
   - `git commit -m "My first commit!"` This creates a commit
   - `git push` This should upload your changes to GitHub.

   If you now go back to your project's page on GitHub, it should show you the commit you created. Email me the link to this webpage, and this will count as your submission.

7. You will find two files in this directory:

   - `assignment1sub.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.
   - `assignment1tests.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.

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

```
#use "assignment1hw.ml;;"
#use "assignment1tests.ml;;"
```

   You should be able to use auto-completion.

   - The first `#use` should print for you the type signatures for all the functions you had to write. Make sure this matches the signatures described in the code file.
   - The second `#use` should print a bunch of `true` values out, for all the existing tests along with all the tests you added.