Computing Science and Mathematics
CSCU9Y4 Practical 9 Functional Programming in OCaml!

0. Our OCaml environment is going to be online and in-browser. There are two online compilers (feedback on either or both is welcomed),

   a) https://try.ocamlpro.com -- great to run through the tutorial! Multi-line programs, however, should be written externally in a text editor, then dragged into the environment.
   b) https://ocsigen.org/js_of_ocaml/2.7/files/toplevel/index.html -- the in-environment editing is a little nicer (eg. ctrl-enter instead of shift-enter for multi-line code). It also translates OCaml into javascript. How cool is that?
   c) https://www.tutorialspoint.com/compile_ocaml_online.php -- requires compilation before execution, much as a local environment. Editing overall is easier, though the UI may be a bit 'finnicky.'

3. Coming from a 'C' world, there are a few things worth keeping in mind:

   - A C-like call, `repeat("hello", 3)` says "call the function repeat with *two* arguments, with first argument as string and second argument as a number."
   - The equivalent OCaml call is `repeat "hello" 3` where no parenthesis appear.

   In OCaml, *parentheses enclose functions*. So the above call could, in its entirety, be enclosed in parenthesis. Since the function call is 'top-level', the enclosing parentheses can be omitted. Consider the following two generic calls to function `f`; note the comments:

```
f 5 (g "Yes!") 3  (* f has three argments; g has one argument *)

f 5 (g "Yes!" 3)  (* Both f and g have two arguments each *)

f (g 3 4)         (* f has one argument, while g has two *)
```

   Recall that in a functional languages, *functions and expressions are the first class components*.

4. The use of **`;;`** and **`;`** and nothing at all.
   - **`;;`** denotes top-level statements, i.e. function defintions. These should *never* be used inside a function or expression.
   - **`;`** is less common, with a meaning that is identical to C/Java. It says, "make sure to execute the code on the left side, before proceeding to the right (or any other code that follows). Rarely will you see use of a single semi-colon
   - Most often, **lines end with nothing at al**l.

5. `Pervasives` is the name of the default library that is loaded in any OCaml environment. See,

   - https://caml.inria.fr/pub/docs/manual-ocaml/libref/Pervasives.html

For example, scroll down to "Pair Operations." That should make clear where the predefined functions come from in Lesson 1, Step 6 of the Try OCaml lessons.

6. Work through lessons 1 through 3 at https://try.ocamlpro.com -- use any or all of the editors listed above. The goal is to get to anonymous functions and iterators. Only then does the power of functional programming begin to become apparent.

   NOTE: If at all unclear about syntax, compiler feedback, etc, make sure to ask!

Checkpoint

Write a function that squares every number in a list of integers.

Additionally, demonstrators may pose additional questions to ensure your understanding. Make sure to ask questions in return!

Optional Extras:

Write a function that returns the sum of two floating point values. Call your function to ensure that it works as expected. Did it work? If not, do some research online to find out why, and fix the problem(s).