Runing the exam project

The project contains three files `Program.fs` (executable) and `Exam.fs` (the file that you hand in) and `ExamInteractive.fsx` (a script file to load the exam into the interactive environment). The file `Program.fs` just contains a skeleton for printing the results of your function to the terminal, you will need to add your own test cases, and the file `Exam.fs` contains the exam itself and this is the file you should be handing in - you should not hand in the entire project or `Program.fs` or `ExamInteractive.fsx`.

You can either use the debug output functionality provided by the project, or you can use `F#` interactive.

For Questions 4.3 and 4.4 you will need to run your program as a project -- `F#` interactive unfortunately does not work reliably when working with IO. For the rest of the project, however, either approach works.

Running the program from the command line

`Program.fs` contains functions where you can write debug outputs for all of your functions. For instance, it contains a function `testQ1 : unit -> unit` where you can place debug printing information (it already contains these for Q1.1, and you can debug other functions in a similar way).

To work in this way all you nead to do is.

1. Open the project
2. Add your solutions to `Exam.fs`
3. Add debug-printing to `Program.fs`
4. Run the project

Running the program from F# interactive

To work in `F#` you have two options

1. change the first line in `Exam.fs` from `module Exam2020` to `module Exam2020 =`. The file will now load as is into the interactive environment (and you can open the module with `open Exam2020;;` in the interactive environment if you wish), although the project will no longer compile. To switch between the two modes, just remove the `=` to go back to project mode. It will work immediately. For the submission the `=` should ideally not be there, but we are not going to dock points if you forget to put it back.

   Another option is to remove the line alltogether. This is, however, dangerous as you may very well get indentation problems when you put it back in. By keeping the module you can safely switch between project mode and interactive mode without worrying about anything breaking.

2. The file `ExamInteractive.fsx` contains only the following two lines:
   ```
   # load "Exam.fs"
   open Exam2020;;
   ```
   To load your solution into `F#` interactive

   1. make sure you have saved `Exam.fs`
   2. open `ExamInteractive.fsx`,
   3. select the entire file and type `Alt-Enter` in VS. Note that just cut-and-pasting these lines rarely works unless you provide the entire path in the `load` command. Depending on the IDE it may complain that it does not find the file, if that is so, input the entire path into the `load` statement.

   From here you can run your functions in the usual way.

If you cannot load the project

If you for some reason cannot load the project then create your own and import `ExamInteractive.fsx` , `Exam2020.fs` , and `Program.fs` in that order.