COMP 3958: Lab 3

For this lab, you may use the functions in the standard OCaml library, but not from other libraries. Submit a zip file containing the two files: `sum.ml` and `records.ml`. Your files must compile; otherwise you may receive no credit for this lab. (We'll be using the `ocamlbuild` command to compile them.). Note that you may be asked to explain your program. Maximum score: 15

1. (a) Implement a function `words` that keeps reading words from standard input until end-of-file and returns a list of the words read. Its signature is

   ```
   val words : unit -> string list
   ```

   Test your function using I/O redirection. Note that words are separated by whitespace.

   (b) Using `words`, implement a program that sums integers read from standard input (until end-of-file). This version skips any word that is not an integer. The program prints the sum when it finishes.

   Put your `words` function together with the code for part (b) in a file named `sum.ml`.

2. Write a stand-alone Ocaml program to sort records read from a file (whose name is specified on the command-line) and display the sorted list of records.

   Each line in the input file contains information of at most one record. The following shows 3 line in a sample file:

   ```
   homer simpson 25  # bad at nuclear engineering
   ned flanders 99  ! high score
   monty burns
   ```

   For a line to be a valid record, it must have at least 3 words and the third word must be an integer between 0 and 100 inclusive. The first word is regarded as the first name, the second word the last name, and the third word is the score. Any extra text after those 3 words is a comment. Your program needs to make sure that a line it reads contains at least 3 words and that the third word is an integer between 0 and 100 inclusive. If not, that line is invalid and is skipped.

   The sorted output should be in descending order of scores, and if multiple records have the same score, they are then in ascending order of last names. Finally, records with the same score and last name are in ascending order of their first names. The following shows the output format for the record in the first sample line above:

   ```
   25 simpson homer
   ```

   Note that the output starts with the score followed by the last name, and then by the first name. Comments are not displayed in the output. Sorted records are displayed to standard output.

   Name your source file `records.ml`. Appropriate error-handling should be performed if a file is not specified on the command-line or if the specified file cannot be read. Note that the program only needs to handle one file.