

MATHFUN

Discrete Mathematics and Functional Programming

Worksheet 9: Input/Output II

Introduction

This worksheet aims to provide you with further practice using Haskell's input/output mechanism as preparation for the functional programming assignment.

You will write a program that maintains a list of words in a file, and allows the user to add words, to display all words, and to display all words of a given length.

Functional Code

Begin by writing three functions:

1. A function to add a string to the end of a list:

```
addWord :: String -> [String] -> [String]
```

E.g., `addWord "lemon" ["apple","banana"]` gives `["apple","banana","lemon"]`.

2. A function to turn a list of strings into a multi-line string:

```
wordsToString :: [String] -> String
```

E.g., `wordsToString ["apple","banana"]` gives the string:

```
"apple\nbanana"
```

which, when output using `putStrLn` will display the words on two lines.

3. A function to give all words of a given length:

```
wordsOfLength :: Int -> [String] -> [String]
```

E.g., `wordsOfLength 5 ["apple","banana","lemon"]` = `["apple","lemon"]`.

User Interface Code

Create a textfile `words.txt` which contains an initial list of words:

```
["apple"]
```

Now, write a program (i.e. a main function) which performs the following steps in order:

- using `readFile` and `read`, reads the contents of `words.txt` (as a single string) and turns this string into a list of strings (see the example use of `read` in the `getInt` function in lecture FP8/9).
- using `addWord`, adds the word “lemon” to the list
- using `wordsToString`, displays the words on the screen
- using `show` and `writeFile`, turns the list into a single string and writes it back to `words.txt`.

Now, modify your program so that it performs the first step (reading the list in from the File) when it starts, and then provides a menu to the user with the following options:

- add a word to the list
- display all words
- display all words of a given length
- exit

On exiting the program, the list should be written back to the file.