Functional Programming Skills
Assignment 4

Arthur Nunes-Harwitt

Explicitly write the type of each function.

1. (24 points) Recall the function `index` from the first assignment. We will write it again in several different ways. Your functions should be structurally recursive and they should *not* make use of accumulative recursion. The types will be essentially the same as the original; the type of the returned value is `Maybe Int`. The time complexity should be $\mathcal{O}(n)$.

   (a) Use the fact that `Maybe` is an instance of `Functor` to write the function `index_a`; take advantage of the fact that we're adding one each time and use `fmap` on the successor function.
   Example:
   ```
   index_a 'x' "qrsxyz" ⇝ Just 3
   index_a 'x' "qrsyz" ⇝ Nothing
   ```

   (b) Use the fact that `Maybe` is an instance of `Applicative` to write the function `index_b`. Make sure to use the `Applicative` operators `pure` and `<*>`.
   Example:
   ```
   index_b 'x' "qrsxyz" ⇝ Just 3
   index_b 'x' "qrsyz" ⇝ Nothing
   ```

   (c) Use continuation passing style to write the function `index_c`. Take advantage of the ability to throw away the continuation when encountering the empty list.
   Example:
   ```
   index_c 'x' "qrsxyz" id ⇝ Just 3
   index_c 'x' "qrsyz" id ⇝ Nothing
   ```

   (d) Use the continuation monad to write a function `index_d`. Take advantage of the `abortWith` operator when encountering the empty list. Also write a function `topIndex_d` that calls `index_d` and provides the initial continuation.
   Example:
   ```
   (index_d 'x' "qrsxyz") <<< id ⇝ Just 3
   (index_d 'x' "qrsyz") <<< id ⇝ Nothing
   topIndex_d 'x' "qrsxyz" ⇝ Just 3
   topIndex_d 'x' "qrsyz" ⇝ Nothing
   ```

(e) It turns out that the `Maybe` type is an instance of `Monad`. Use the return and bind operators (but *not* `do` notation) to write the function `index_e`.
Example:
```
index_e 'x' "qrsxyz" ⇝ Just 3
index_e 'x' "qrsyz ⇝ Nothing
```

(f) Finally, use the return and `do` notation to write the function `index_f`.
Example:
```
index_f 'x' "qrsxyz" ⇝ Just 3
index_f 'x' "qrsyz ⇝ Nothing
```

2. (5 points) Write a function `meetAndGreet`. When invoked, a message is displayed requesting the user's name. Once the name is entered, a message is displayed to greet the user by name.
Example:
```
*Assign4> meetAndGreet
What is your name?  Dolly
Hello Dolly!
```

3. (20 points) Write a function `average` to compute the mean of a list of doubles. Then write the function `readDoubles` that takes two strings: a prompt and a sentinel. It is used to read in a list of doubles. Write another function `interface` that displays instructions, reads in a list, computes the mean, maximum, and minimum of the list, and then displays those results.
Example:
```
*Assign4> interface
Enter some numbers.
When finished, type 'done'.
Enter a number:  2
Enter a number:  1
Enter a number:  3
Enter a number:  done
The average is 2.0
The maximum is 3.0
The minimum is 1.0
```

Graduate Problems/Undergraduate Extra Credit

1. (5 points) Write a function `cp` that takes two strings representing file names (and paths) and copies the contents of the first file, giving the copy the second file name.