COMP 3007 A/B (Winter 2023) – "Programming Paradigms"

Carleton University

Specification for Assignment 5

***Your submission** for this assignment **must be commented** and **must include both your name** and your **student number** as a comment at the top of **every source file you submit**. Each of your submitted files must use a **file name beginning 'comp3007_w23_#########_assignment_05'** (replacing the number signs with your own student number) and any **submissions that crash** (i.e., terminate with an error) on execution will automatically **receive a mark of 0**.*

Officially, the Due Date for this Assignment is:

***Friday, March 10th, 2021**, at **11:59pm EST**.*

Late Submissions are Accepted Without Penalty Until Sunday, March 12th, by 11:59pm EST.
Submissions received after that will not be accepted and will receive a mark of 0.

The objective of this assignment is allow you to practice with list processing and custom "database-like" definitions in Haskell by designing and implementing a program for retrieving the phone numbers from a directory that are associated with a name that "sounds like" a name specified as an argument.

The "Soundex" algorithm was developed to support search operations where the precise spelling of the key value was less important than the phonetic pronunciation. If, for instance, one was searching a database for the name John, the name Jon would also be considered a match. Soundex codes are strings of exactly four characters in length, created by the following algorithm:

1. append the first character of the word to the (currently empty) Soundex code
2. remove any and all occurrences of the letters h or w that occur after the first character
3. remove any duplicate letters
4. remove all occurrences of the letters a, e, i, o, u, or y that occur after the first character
5. convert every character after the first to a numerical digit, using the following code:

B, F, P, V	1
C, G, J, K, Q, S, X, Z	2

D, T	3
L	4

M, N	5
R	6

As a clarifying example of the Soundex algorithm, a last name of COLLIER would first undergo the transformation COLLIER → COLIER → CLR → C46, and the resulting Soundex code would be C460. This would "match" the name COLYER, but not COLLIE.

For this assignment:

- you must design and implement a series of functions that are capable of producing the Soundex code associated with a name (according to the procedure provided above).
- you must define (mirroring the video rental database type presented in class) a list of tuples (pairs) wherein the first element will be a last name (written in uppercase) and the second element will be a phone number (stored as a 12-digit string with hyphens at indices 3 and 7; e.g., "613-520-2600").
- you must create a "query" function defined such that when it receives a database (i.e., a list of tuples of names and phone numbers) as its first argument and a name as its second argument, it produces a list containing every phone number in the database associated with a name that has the same Soundex code as the second argument. Although you are expected to test your code thoroughly, please note that the teaching assistants will be provided with their own database for testing.