Homework #02

Complete By: 10/08/2023 @11:59pm
Policy: Individual work only, late work *not* accepted
Assignment: complete F# program
Submission: submit "main.fs" electronically on Gradescope

Pre-requisites: Lecture information

Reading / F# References

Here are some references on F# that prior students have found helpful:

http://dungpa.github.io/fsharp-cheatsheet/

https://fsharp.org/learn.html

https://www.tutorialspoint.com/fsharp/index.htm

Programming environments for F#

The quickest way to get started is using replit.com: there is a project setup for this homework named "**Homework 02**" which is configured with F# and starter code. Alternatively, you can install Visual Studio Community Edition for Windows or Mac, both offer F#; create a new F# Console project to start programming. If you decide to work outside replit.com, be sure to download the starter code of "main.fs".

[*NOTE: Visual Studio Code is a different product, and is an editor, not an IDE --- you can use VS Code if you want, but you'll also need to install the necessary F# libraries and compilers.*]

Homework Exercise

The exercise is to input a string from the user, and then output the following information:

- Length of string (# of characters, including spaces and punctuation)
- # of high frequency letters in the string (e, t, a, o, i, n, s, r, h, l)
- # of each high frequency letter

- The result of making the following substitutions for words in the input string
  - “the” to “her”
  - “boy” to “man”
  - “rat” to “hat”

  These substitutions may be within words. For example, them -> herm and boyboy -> manman

For simplicity, **only count lower-case letters**. To help you get started, we are providing starter code that inputs a string from the user and calls a few functions that currently return a zero:

```
> dotnet run
Starting

input> this is f sharp
exploded: ['t'; 'h'; 'i'; 's'; ' '; 'i'; 's'; ' '; 'f'; ' '; 's'; 'h'; 'a'; 'r'; 'p']

length of sentence: 0
# of top 10 letters: 0
imploded: "this is f sharp"

Done
>
```

When your program is complete, here’s an example of how it should behave given the input: **a boy is running**

```
> dotnet run
Starting

input> a boy is running
exploded: ['a'; ' '; 'b'; 'o'; 'y'; ' '; 'i'; 's'; ' '; 'r'; 'u'; 'n'; 'n'; 'i'; 'n'; 'g']

length of sentence: 16
# of top 10 letters: 9
'a': 1
'e': 0
'h': 0
'i': 2
'l': 0
'o': 1
'n': 3
'r': 1
's': 1
't': 0
imploded: "a boy is running"

swap imploded: "a man is running"

Done
>
```

The most natural way to solve this problem in a functional language is to convert the input string to a list of characters, and then work with the resulting list. Two functions are provided for converting strings: **explode s** takes a string and returns a list of characters, and **implode L** takes a list of characters and returns a string. Here’s the starter code here for your reference:

```
//
// F# program to input a string and print out information
// about the # of top 10 letters in that string and substituting certain words
//
// Name: ???
// NetID: ???
// UIN: ???

// Original Author:
//  Prof Jon Solworth
//  U. of Illinois, Chicago
//  CS 341, Fall 2023
//


//
// explode:
//
// Given a string s, explodes the string into a list of characters.
// Example: explode "apple" => ['a';'p';'p';'l';'e']
//
let explode (S:string) =
  List.ofArray (S.ToCharArray())

//
// implode
//
// The opposite of explode --- given a list of characters, returns
// the list as a string. Example: implode ['t';'h';'e'] => "the"
//

let implode (L:char list) =
  new string(List.toArray L)

//
// returns length of the string
//
let rec length L =
  0

//
// returns # of top 10 letters of the english alphabetin the string
//
let rec topTen L =
  0



[<EntryPoint>]
let main argv =
  printfn "Starting"
```

```
printfn ""

printf("input> ")
let input = System.Console.ReadLine()

let L = explode input
printfn "exploded: %A" L

printfn ""
let len = length L
printfn "length of sentence: %A" len

let num = topTen L
printfn "# of top 10 letters: %A" num

//
// TODO: print count for each of the top 10 letters
//

printfn ""
let S = implode L
printfn "imploded: %A" S
printfn ""

//
// TODO: substitute a series of letters from a given string
//

//
// TODO: print updated sentence
//

printfn ""
printfn "Done"
0 // return 0 => success, much like C++
```

When producing output, use **printfn** with the **%A** option as shown above. This is the easiest way to output values in F#, and will match the required output when submitting to Gradescope.

Requirements

In order to earn full points for this homework, your code must meet the following requirements:

- No variables (i.e. no use of the keyword mutable).
- No loops and no if-else. Instead, use recursion or higher-order programming via List.map or List.iter.
- Compute the length of the list yourself, do not call List.length; recall we wrote the length function in class.

Challenge: it's easy to fall into the trap of relying on copy-paste to solve this exercise. For example, you can write a function to count the number of 'a', then copy-paste and modify to count the number of 'e', then copy-paste again to count the number of 'i'. And so on. Instead, write a single function that is parameterized by the top 10 letter, and just call this same function 10 times. Then, the next step would be abstract those 10 calls into one call to List.iter / List.map.

Grading and Electronic Submission

A few days before the assignment is due, login to Gradescope.com and look for the assignment "Homework 2". Submit your F# program file "main.fs" --- you have unlimited submissions. Keep in mind we grade your LAST SUBMISSION unless you select an earlier submission for grading. You need to complete all the functions in order to get credit for this assignment.

Also, add your name to the header comment, and comment any functions you write!

Academic Conduct Policy

Late work is not accepted for this assignment. All work is to be done individually — group work is not allowed. While we encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you *cannot* work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own. The University's policy is available here:

https://dos.uic.edu/conductforstudents.shtml

In particular, note that you are guilty of academic dishonesty if you **extend or receive any kind of unauthorized assistance**. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums (e.g. you cannot download answers from Chegg). Other examples of academic dishonesty include emailing your program to another student, sharing your screen so that another student may copy your work, copying-pasting code from the internet, working together in a group, and allowing a tutor, TA, or another individual to write an answer for you. Academic dishonesty is unacceptable, and penalties range from a letter grade drop to expulsion from the university; cases are handled via the official student conduct process described at https://dos.uic.edu/conductforstudents.shtml .