**CSCE2014 Programming Foundations II Homework One Project Report**

In this homework project, I learned more about the applications of helper files in using functions from a separate cpp file, the relation between pointers and arrays + their usage in for loops, and the effects of ‘\0’. The coding of functions like existing library functions helped me understand some C++ framework and a deeper knowledge of scope and testing. The biggest takeaway from the testing portion of this project was the sheer amount of cases a function must be prepared for and how to correct my existing code for unexpected errors. Overall, the assignment successfully improved my knowledge of function implementation, null character situations, and arrays in C++.

The reason for having so many varying test inputs and predicted outputs is to cover a wider range of subsets of the function being tested to see if it properly runs as expected.

*\*F1 = Function 1 (bool is\_char\_in\_the\_set(char c, const char \*set))*

*\*F2 = Function 2 (char \*find\_first\_not\_in\_the\_set(char \*str, const char \*set))*

*\*F3 = Function 3 (char \*find\_first\_in\_the\_set(char \*str, const char \*set))*

**Initial Sample Test:**

* **String= “Hi Everyone!” , Set= “e”** : tests a true case for *F1* and a set size of 1 for *F2* and *F3*
  + F1: True, F2: “e”, F3: “H”

**My Tests:**

* **String= “Corona virus?” , Set= “covid-19”** : tests a set size of >1 for *F2* and *F3*
  + F2: “C”, F3: “o”
* **String= “Ahmed”, Set= “Ahmed”** : tests a case w/ equal string & set (show a null output)
  + F2: NULL, F3: “A”
* **String= "1 29 )&@ 721 =-92!3\*\*#&", Set= "@#)\_(\* \*&^5 7 -- 75 (((001000"** : tests set size > string size and symbols + spaces
  + F2: 2, F3: 1

**Sample Testing of 1st Function:**

*Se = “abcd”*

* **X = ‘a’ and X = ‘d’** : tests a true case (a is in abcd and d is in abcd)
* **X = ‘#**’ : tests a false case (# is not in abcd)
* **X = ‘\0’** : tests a true case (null character input) to show that the set ends with ‘\0’

*Test\_set[0] = ‘\0’*

* **X = ‘a’** : tests a false case (a was replaced with a null character)
* **X = ‘\0’** : tests a true case (null character input into changed set)

**Sample Testing of 2nd Function:**

* **String= "ls -l -a | wc -c >> myfile", Set= “c”** : tests a set size of 1 (1st in String is not in Set)
  + Output: address l, cstring "ls -l -a | wc -c >> myfile"
* **String= "ls -l -a | wc -c >> myfile", Set= “sl -c”** : tests a set with a size >1 (1st not in set is in the middle of the string, not the beginning)
  + Output: address a, cstring "a | wc -c >> myfile"
* **String= "-- // --- -", Set= " -/"** : tests a set size where all characters of the string are in Set (checks if nullptr output works)
  + Output: NULL
* **String= "-- // --- -", Set= "abc” :** tests a set where no characters of the string are in Set
  + Output: address -, cstring "-- // --- -"
* **String= "ls -l -a | wc -c >> myfile", Set= “ls-l-a|wc-c>>myfile”** : tests a set where the string and set are the same except for spaces (checks if F2 will recognize & output the address of a space)
  + Output: address , cstring " -l -a | wc -c >> myfile "
* **String= "ls -l -a | wc -c >> myfile", Set= “ls -l -a | wc -c >> myfilE”** : tests a set where the string and set are the same except for an uppercase letter (checks if F2 will recognize the difference in case)
  + Output: address e, cstring "e"
* **String= "ls -l -a | wc -c >> myfile", Set= “”** : tests an empty set/null input (the 1st letter can’t be in an empty set, thus it is the output of F2)
  + Output: address l, cstring "ls -l -a | wc -c >> myfile"
* **String= "", Set= “ xze”** : tests an empty string/null input (there can’t be a 1st not in set if the string is empty, thus the output of F2 is null)
  + Output: NULL

**Sample Testing of 3rd Function:**

* **String= "ls -l -a | wc -c >> myfile", Set= “c”** : tests a set size of 1 (1st in set is in middle of string)
  + Output: address c, cstring "c -c >> myfile"
* **String= "ls -l -a | wc -c >> myfile", Set= “sl -c”** : tests a set with a size >1 (1st in set is at the start)
  + Output: address l, cstring "ls -l -a | wc -c >> myfile"
* **String= "-- // --- -", Set= " -/"** : tests a set size where all characters of the string are in Set (tests if output is the 1st that is in the set)
  + Output: address -, cstring "-- // --- -"
* **String= "-- // --- -", Set= "abc” :** tests a set where no characters of the string are in Set (checks if nullptr works)
  + Output: NULL
* **String= "ls -l -a | wc -c >> myfile", Set= “bdgEe”** : tests a set where the string and set are different except for one letter (e at the end; should loop through the full string & set)
  + Output: address e, cstring "e"
* **String= "ls -l -a | wc -c >> myfile", Set= “”** : tests an empty set/null input (characters of the string cannot exist in an empty set, thus the output of F2 is null)
  + Output: NULL
* **String= "", Set= “ xze”** : tests an empty string/null input (there can’t be a 1st in set if the string is empty, thus the output of F2 is null)
  + Output: NULL