## **CSX-351 Homework 3**

## Dynamic memory, searching and strings.

The purpose of this assignment is to give you practice using dynamic memory allocation, string functions, sorting, and searching. You will write a program to check keyword occurrence and test it.

### **Program Steps**

1. Read an unsorted keywords file once to determine how many words are in the file.
2. Allocate memory dynamically to store the unsorted keywords in an array of strings.
3. Reread the keywords file a second time and store the words in the dynamically allocated array of strings.
4. Sort the array of key words. (Hint: be sure to check your sorted array of key words - there should be 84)
5. Search a C++ program input file for occurrences of the keywords:
   * Read one line at a time from the code file.
   * Parse each line into separate words. **Ignore any words that are inside a comment.**
   * Search the keyword array to determine if each word is a keyword.
   * For keywords, print the line number, the keyword, and the position of the keyword in the line.
   * Keep a count of the number of keywords found

### **Program Requirements**

* Make sure you check the input file for successful opens.
* Your output should match the format show below with the correct line number and position of each word in the line. The line character positions start at zero. Note, there are more than 50 lines of output.

### **Program Output**

Your output should looks like this:

|  |
| --- |
| Line 8: using(0) namespace(6) **<== *using* occurs at position 0 on line 8, *namespace* occurs at position 6 on line 8**  Line 10: const(0) int(6)  Line 12: void(0) const(19)  Line 13: void(0) char(20) int(32) const(48)  Line 14: bool(0) const(24) char(30) const(42)  Line 15: void(0) char(17)  Line 16: void(0)  Line 17: void(0)  Line 19: int(0)  Line 21: const(4)  ...  Number of keywords found = ??  **<== Add this line at the end of your output, replace ?? with the correct number** |

#### **The keyword file looks like this:**

|  |
| --- |
| for  if  nullptr  break  int  long  sizeof  return  short  else  friend  const  static\_cast  ... |

#### **The input code file looks like this:**

|  |
| --- |
| #include <cstdlib>  #include <cstring>  #include <cctype>  #include <cmath>  #include <fstream>  #include <iostream>  #include <string>  using namespace std;  const int DictionarySize = 23907;  void getDictionary(const string& filename,string\*);  void spellCheckLine(char\* line, int lineNumber, const string\* dictionary);  bool wordIsInDictionary(const char\* word, const string\* dictionary);  void toLowerCase(char\* text);  ... |