CSCI-311 Assignment 4

Overview:

1. Command line arguments.
2. Read from a file and write to a file.
3. How to test a program that has *.*out* and *.*cmd* files.
4. How to measure time in seconds inside a C++ program.
5. ASCII character representation and how to convert characters to lower case in constant time.

To get credit for this assignment, you need to write four functions:

1. Function ***readFromFile*** will take as a parameter an empty string S by reference, and a string, name of a file. This function will open the given file, and will read all the lines from the file and concatenate the lines into S. At the end of the function, S will be the concatenation of all lines from the given file.
   void *readFromFile*(string &S, string filename)

2. Function ***convertToLower*** will take a string S passed by reference as a parameter and will convert all the characters in S to lower case (you may use std function ***tolower***).
   void *convertToLower*(string &S)

3. Function ***lessThan*** will take as parameters a constant string passed by reference *S*, and two integers, *first* and *second*, the starting indices of two suffixes in *S*. This function will compare the two suffixes of S starting at their starting positions and to the end of S one character at a time. The function will return true if the suffix starting at *first* is less than the suffix starting at position *second.*
   IMPORTANT: do not use ***substr*** to extract suffixes from the given string; do not copy suffixes into a separate strings – this takes O(n) time and O(n) space. We would like to save time and space.
   bool *lessThan*(const string &S, int first, int second)

4. Function ***partition*** that takes as parameters (1) string ***S*** of size *n*, passed constant by reference (2) vector ***indices*** of integers of size *n*, where each integer is the starting position of a suffix in S; *indices* are passed by reference (3) integers ***low*** and ***high,*** the start and the end indices of a given range; and (4) an integer ***pivotIndex***, which is an arbitrary index of the vector *indices*.
   int *partition*(const string &S, vector<int> &indices, int low, int high, int pivotIndex)

   - ✓ First, this function swaps *indices*[*pivotIndex*] and *indices*[*high*], where *high* is the last index in the range.
   - ✓ Then, this function will partition suffix indices (use code of Partition provided in lecture notes as an example) so that the first half contains positions of suffixes that are less than suffix *pivot* and the second contains indices of suffixes that are greater than suffix *pivot.* This function will call the functions *lessThan* to accomplish this task.

Inside your main(int argc, char* argv[])

1. Call *readFromFile* to read S from the file.
2. Call *convertToLower* to convert characters of S to lower case.
3. If argc is equal to 2 (*e.g.*, command line is: ***./main text1.txt***), then print out S using *cout* and *endl* at the end.
4. If argc is 4 (*e.g.*, command line is: ***./main text1.txt 10 20***), then print out S using *cout* and *endl* at the end, and call function ***lessThan*** on indices 10 and 20 (from the command line arguments), if function returns true, print out "True."; otherwise, print out "False.", and *endl* at the end.
5. If argc is 3 (*e.g.,* command line is: ***./main text1.txt 30***), then use the argument 30 as a pivot index to pass to function *partition*, call partition on S and this pivot index and print out the resulting *indices*, with a space after each index, and *endl* at the end.
6. If argc is not 2, 3 or 4, return -1, and do nothing.

To test your program, use test files inside tests directory inside Assignment5.
To untar Assignment5.tar, use the command:
tar -xvf Lab4.tar
Your program will use command line arguments stored inside *.*cmd* files and input files inside Assignment5.

Example of the content of t01.cmd:

```
text1.txt 10 20
```

This means that you need to run your executable using this command:
./main ***text1.txt*** **10 20** > *t01.my*

This will use **text1.txt**, **10** and **20** as the command line arguments and will redirect the output from your program into *t01.my* file. To compare your output and the correct output, use:
diff t01.my tests/t01.out
vimdiff t01.my tests/t01.out

Submission: submit ***main.cpp*** to Assignment5 to turnin.

Grading: If your program does not compile, you will receive 0. The rest of the grading will be done as shown in the table below.

Function	Points	Test files
readFromFile	15	t00, t01, t02
convertToLower	10	t01, t02
lessThan	25	t03, t04, t05
partition	50	t06, t07, t08, t09