**Notes:**

- All OutputFiles are under the folder “./OutputFiles”  
- More detailed changes are commented in the .cpp file

**Changes:**

- Add an Output folder.

- Under IsFirstAboveSecond() the return statement should be "return (i == \_first.length())"

- In the DoSingleThreaded() function, the ReadFile is modified, the method of returning all strings from looping is abandoned, and the time complexity is reduced by fetching the address

- Use QuickSort() for better time performance

- Change the ReadFile() function to fix the bug of not getting the last string in file, and the list is returned directly by reference now

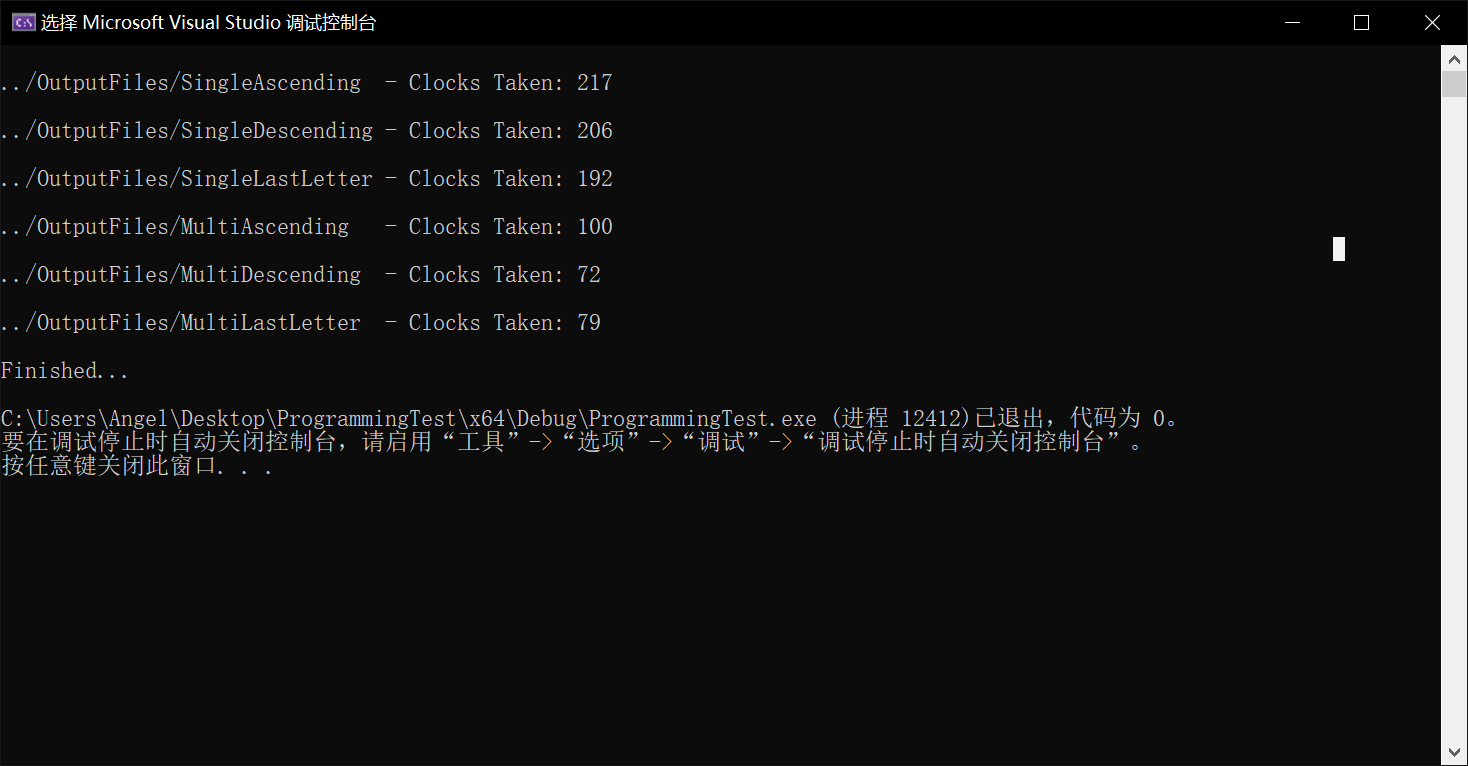
- DoMultiThreaded() function involves a total of 4 files, so "\_fileList.size()"does not need "-1" to create 4 threads. "join" should be placed in a loop to enable multi-threading

- Using the RAII technique (Resource Acquisition Is Initialization), we should not call "file.close()", because if an exception is thrown it may not close

- Change some parameters to const reference

**Result:**

Significantly reduce time consumption. Multithreaded method use less time than Singlethreaded one in general.



**Recommended Future Changes:**

- Change string Vector to string Set to deduplicate. However, if one word has 2 meanings, it seems reasonable to put it as 2.

- Case sensitivity. If necessary, you change to all lowercase or uppercase in advance and then deduplicate. (See above for the deduplication method)

- Empty strings (spaces) will also be read as strings. Is this reasonable? If not, trim it. Spaces (or any other illegal char) in a string can also be trimmed like this:

std::string\_view ltrim(std::string\_view s)

{

s.remove\_prefix(std::distance(s.cbegin(), std::find\_if(s.cbegin(), s.cend(),

[](int c) {return !std::isspace(c);})));

return s;

}

std::string\_view rtrim(std::string\_view s)

{

s.remove\_suffix(std::distance(s.crbegin(), std::find\_if(s.crbegin(), s.crend(),

[](int c) {return !std::isspace(c);})));

return s;

}

std::string\_view trim(std::string\_view s)

{

return ltrim(rtrim(s));

}

- Also apply concurrency feature to the “sorting” part. However, this depends on how many data that need to be sorted, e.g., if the number of string is larger than 5000, use multithreaded sorting method, otherwise do it normallly.