**CM2204 Advanced Programming – C Programming Coursework Results**

**Student: Mohammed Kashem**

**Functionality:**
Read two documents from ASCII files (first/second argument): OK
Identify words in ASCII file: OK, but word length limit not ideal.
Detect words present at least once in both documents: OK
Order words in increasing alphabetical order: OK
Write ordered list of words on the terminal: OK
Process error conditions : OK

*Excellent:* efficient and complete implementation of all required features

**Design & Structure:**
Suitable processing of input file contents: OK
Suitable data structure/algorithm for identifying common words: OK
Suitable data structure/algorithm for sorting words: OK
Suitable approach toward printing results on terminal: OK
Overall clearly structured code: OK

*Excellent:* well structured and concise code; optimal use of suitable data structures and algorithms

**Documentation:**
Comments reveal structure of code: OK
Concise comments: OK
Comments discuss high-level idea of approach: OK
Comments clearly explain complicated parts of the code: OK

*Excellent:* useful comments clarifying the main ideas and high-level structure in a concise way

Really well done this time. You could still improve this by using a single binary search tree, add words for the first file and mark words from the second file in that tree as being present, also using a balanced tree would make things even faster, but these are rather advanced options.

| | | |
|---|---|---|
| 90-100: 3x excellent<br>80-89: 2x excellent, 1x good<br>70-79: 1x excellent, 2x good<br>60-69: 2x good, 1x adequate | 50-59: 1x good, 1x adequate<br>40-49: 2x adequate<br>20-30: 1x adequate<br>0-19: Otherwise | **Marks: 95/100** |