classwork 4

task 1

There are N files with .txt extension in a directory. The program allows a user to enter a word or a phrase and the program searches it in all files of the directory. The number of initiated threads should be the number of cores (M). The results should be written to another new text file as the word is found. For example, the files are A.txt, B.txt, C.txt, D.txt, E.txt, F.txt and G.txt. The user enters “capital”. The program finds it in B.txt and then in C.txt after that in A.txt and D.txt. etc. So the results.txt may be something like:

```
word to find: “capital”
thread id: “2” B.txt line 2
thread id: “3” C.txt line 7
thread id: “1” A.txt line 38
thread id: “4” D.txt line 29
thread id: “3” E.txt line 15
thread id: “2” F.txt line 64
thread id: “1” G.txt line 99
There are 7 matches found
```

Here in file A.txt the word was found a little later than in D.txt, but it took the queue to write to the results.txt earlier. It is fine if line numbers which are close to each other are not in the right order.

Use multithreading for searching and mutex for writing the result. Pay attention that the number of threads is M, unlike two in the producer and consumer example. So, use the relevant notify function. Keep updating the number of the matches in one variable using the C++ tools like mutex and conditional variable to avoid race conditions and write it in the end.

Submit firstName_lastName.cpp file.