Question 1

Problem statement: You have given two input text files. Create an output file that is obtained by merging the given input files. Moreover, you have to convert all the uppercase characters in the output file into lowercase characters.

Input format: You have given two input files.

Output format: You have to write the output to a text file.

Explanation: You have been provided two input files named input1_a.txt and input1_b.txt. You have to merge the given files(space as files content separator) and convert all uppercase characters to lower cases. The output is stored in an output file named output1.txt.

Sample input:

input1_a.txt:
Hello

input1_b.txt:
World

output1.txt:
hello world

Execution should happen like this:
./a.out input1_a.txt input1_b.txt output1.txt

Question 2

Problem Statement: Find the number of occurrences of a given word in the file.

-The program should take 3 command line arguments:
1) Input file to read.
2) Word to count frequency.
3) Output file to write the o/p into.

-The output file MUST contain the word and the frequency of the given word.

-You may assume that a word in the input file will be of a maximum length 100.

-The matching should be case insensitive. (i.e, word == WORD)
Example: Search for word "game" and "Game" should give the same results for a given input file.

-A word which is a substring of another big word should not be matched.
Example: While searching for the word "ever", occurance of "everlasting" should not be taken into account.

-You may assume that the text file may contain only ! . " ( ) , ? ' as special characters. In addition, 2 words may be separated by spaces.

-The input word to count frequency doesn't have any special characters.

Example

Sample Input -
Here’s the fact. Why must such a thing even exist? Given a set of (Key, Priority) pairs, how do we know it is even possible to design a tree so that the keys are in search tree order and the priorities are in heap order?

Output - Frequency of key : 1

Explanation: key is matched with Key, but not with keys.

Execution should happen like this:
./a.out input.txt word output.txt