Internship Question

Instructions:

- Duration: 8 Hours (need not to be single sitting)
- You can choose to solve any one of the questions or both of them.
- Copy and paste the question in your own drive, please don't edit on this page.
- Upload the solution on your google drive and submit the link in the form shared with you by the name of Internship challenge (Elitmus).
- Make sure that you keep the solution drive link public.

Problem 1: Iterative Word Suggestion

Given a CSV* file with words and their frequency.
Build an executable file which should take a word, character by character input (only english ASCII characters), and print the 5 nearest matching words with highest frequency and the time taken to get the output in each iteration as well. Exit with **‘#’** sign or when there is no match found.
Output will be according to all the input characters.
executable must be portable, should run on any 32-bit windows machine.
No other dependencies should be there for the executable to run.

Example:

```
k
Know, kiss, kill, kick, keyboard          250  µs
i
Kiss, kill, kick, kite, kitten            200  µs
l
Kill, killer, killed, kilobyte, kilogram, 238  µs
l
Kill, killer, killed, killable            296  µs
q
No match Found !!                         150  µs
Exiting
```

Program Submission Rules:

- There must be 2 directories. All the code should be in a directory named : “**source**” & the executable must be in the directory “**build**” with no other dependencies(not even compiler libraries dependency).
- Code must be compilable. If some additional argument to be given for compilation, please prefer make or Cmake.
- A README file should be there within the “**source**” directory with all the commands in sequence to build the project.

***Link to CSV file** : https://drive.google.com/open?id=12UJI_TjV_JIMVS9XCGLEXfboPXO4IMi3

Problem 2: Swipe to type

Given an input sequence of string(breakpoint sequence) and a dictionary*. Extract the 5 nearest possible words from the string which are present in the dictionary. Character repetition is allowed.
Additional weightage should be given to the frequency of words.

Ex :
INPUT : **ADRE**
OUTPUT : Expected words : Are, Age, Add, Area, Ad

Explanation : In output, the 1st word “Are” can be formed by only removing character ‘D’ and it’s frequency is also very high.
Rest 4 output words can be formed by replacing 2 characters(either by removing or adding/replacing characters)

***Link to dictionary** :
https://drive.google.com/open?id=12UJI_TjV_JIMVS9XCGLEXfboPXO4IMi3