Overall program design

Since I am new to C programing, I strictly follow the assignment instruction, which I believe it provides a clear program design process. I got my ideas to start writing this program from suggested approaches that shows me how to complete the whole program step by step so that I knew how far I was in the whole program. For example, it tells me I should check the validity of arguments first, and then implement opening dictionary and file handing, etc. As a new programmer, following instruction is helpful, I learned how to functionally design program that finishing each functionality step by step and put them together.

Use of dynamically allocated memory

Personally, the biggest challenge in this assignment is using dynamic memory allocation to create a 2d-array to store a filtered dictionary. I got a big trouble in allocating memory and freeing them. After doing some research, I realized the first time I should've allocated char* to each row and then line 1d array for each column, which has a big difference with allocating memory to an 1d array. Using dynamic memory helps me to store what I want in somewhere and print them after sorting in correct order. So, instead of printing each line, I can store all I need and print them all at once.

Use of macros

I find macros helpful. Since macros is preprocessed, I don't need to pay attention to its type. For example, I used macros in my program to define error messages, when I printed them, the program would print exactly what I wanted them to be printed. Also, when I needed to replace a value in my error messages, such as pattern error and dictionary error, I only needed to put %d or %s in where I needed to replace an integer or a string, which was easier to use.

Functional decomposition

In the beginning, I wrote a function that did a lot of things, then I realized it's quite difficult to test when something wrong occurred in this function. Since then, I used one function to only solve on problem, this was helpful when I wanted to test if the function works. For example, when checking validity of arguments, I wrote different functions to check different arguments, and only return true or false to let me know if those arguments were valid or not. This also helped me to trace back when some validity test failed.

Repeated code

I did have some repeated code before my last inspecting. I wrote two functions to sorting alpha and best respectively, later I realized that they were doing the same thing and what I need to do is add another parameter to indicate if it is alpha sorting or best sorting. Because of that, I saved more than ten lines of codes to implement same function.

File handling

To read file and print or save every word, fgets() helps a lot, it reads each line and automatically reads next line until it finds EOF. Thanks to fgets(), what I need to do is store each word that fgets() read for me and get rid of '\n' character before using them.

Error handling

Instead of writing codes to print error messages to stderr and exit the program every time when I need to, I created different functions to handle different errors for me. Firstly, it helps me reduce code redundancy and shows a better structure. Secondly, it's clear to trace back if something wrong is in my error handing.

Testing approach

Apart from provided tests, I tested my code myself sometimes to locate where the problem exists in my code. I tested all my functions before they were in use. What I did was writing some simple input and expected output and call and function in first line of main function and return 0, this helps me test if the function works as I expected before using. Moreover, before putting many functions together, I also tested if they work well together as similar approaches demonstrated above.

What I would have done differently

I did learn a lot in this assignment from my mistakes. I should've tested each function before using them, which could save more time for me to find my bugs. I should've read the instruction more carefully so that I knew all requirements, rather than realized my mistakes and changing all codes that have been written and tested. Moreover, If I get more time, I will try more struct in my code, and try enums. Since those are new concept that I haven't perfectly mastered but useful.