FAQs

1. **Can there be spaces and slashes in the "name of the file" exclusively (not the whole path) that will be given as command-line arguments?**

   "Spaces, and indeed every character except “/” and NULL, are allowed in filenames". Slashes are anyways not allowed in filenames anywhere.
   In case the file name contains spaces, they will be escaped using either quotes or a backslash. Example:

   1. 'test file.txt'
   2. test\ file.txt

   should both work with your program.

2. **Can we make an assumption that the path provided would be only relative or only absolute?**

   File Paths can be both relative or absolute.

3. **Should the reversal work for even formatted pdf and ppt files?**

   It is required for the code to work on normal text files. There are no other requirements. The input files contain only ASCII text, including '\n' and '\0'.

4. **Can we use lseek64 instead of lseek in the assignment?**

   NO, lseek64 is a library routine, not a system call.

5. **What all do we have to include in the README.md file**

   Please mention any assumptions you have made and explain them. If you haven't made any, write that. **Including a README.md is compulsory.**

6. **What is the format to display the percentage of the file written?**

   The percentage can be in the form of the number of characters written. For example: if 50 characters are written out of 1000 characters. Then the progress

shown would be 5%, then 5.01%, and so on. **Percentages should be rounded off to 2 decimals.**

7. **What is the hard limit on RAM in the first question?**

   There is no such limit. You can make a *reasonable* assumption.

8. **Are we allowed to use <math.h>?**

   Use of math.h is not allowed.

9. **For the third question (Part 3) do we have to follow the exact output format specified? Can we instead use the Linux style: `drwxrwxrwx` to print out the permissions?**

   No, the Linux format is not allowed. Please follow the **exact** format specified in the PDF.

10. **In Part 1 and 2, What should we do in case an error is thrown?**

    Try to handle the error in the best way possible.
    In case the error is fatal, such as when the input file does not exist, you do not have much of a choice but to exit the program.
    In case it is possible to skip the affected operation and continue with others, you are free to do that as well.
    In either case, printing a help text highlighting the cause of the error is a good idea.

11. **In Part 3, What should we do in case any input file throws an error? Should we skip permissions for that file or just abort the program?**

    Skip the permissions for the erroneous file, but still, print the output for the remaining program.
    For eg. If the output_file_1 does not exist - print the permissions for the output_file_2 and the directory, display an error message for the output_file_1.

12. **Can we add some extra newlines for indentation of the output?**

Part 1:

- **NO** extra newlines should be added in Part 1. The new file created in part 1 should be the **exact reverse** of the input file.

Part 2:

- **NO** extra newlines should be added in Part 1. The new file created in part 1 should contain the **exact reverse** of the desired block of characters as mentioned in the input.

Part 3:

- You can format the output to look more presentable, but keep the **exact same** contents the same as specified in the PDF.
- You may or may not add a progress bar / percentage. You may include it to enhance your work, but still output the numerical figure (in case you add a progress bar) alongside. Rest is your wish.

13. What to do if the assignment folder already exists?

If the 'Assignment' folder already exists in Part 1 or 2, use the existing folder itself. Overwrite the file inside if it exists as well.