### Project: Directory and File Information System in C

\*\*Description:\*\*

This project is a C-based application that traverses directories, lists files, and retrieves detailed file information such as size, permissions, and modification dates. It allows for optional filtering based on file size, extensions, and running system commands (`ls -l` and `wc -l`) on the listed files. The program utilizes system-level functions like `opendir()`, `readdir()`, `stat()`, and process management with `fork()` and `exec()`.

\*\*Key Features:\*\*

- \*\*Directory Traversal:\*\* Recursively lists files and directories from a specified location.

- \*\*File Information Display:\*\* Outputs file size, read/write/execute permissions, and modification timestamps.

- \*\*Custom Command Execution:\*\* Runs `ls -l` or `wc -l` on files and directories via system calls, with results printed to the console.

- \*\*File Size Filtering:\*\* Filters files based on a minimum size threshold.

- \*\*Recursive Function Calls:\*\* Handles multiple levels of directory nesting.

\*\*Skills Demonstrated:\*\*

- \*\*C Programming:\*\* Demonstrated understanding of system-level programming concepts, including file handling, directory traversal, and memory management.

- \*\*Process Management:\*\* Use of `fork()` and `exec()` for executing shell commands within the program.

- \*\*File I/O and System Calls:\*\* Applied various file I/O operations, including `stat()` for file metadata and `fopen()` for file size calculation.

- \*\*Command-Line Argument Parsing:\*\* Implemented argument parsing to allow users to customize the program's behavior with flags (`-S`, `-s`, `-f`, `-e`).

\*\*Impact:\*\*

This project illustrates strong knowledge in C programming, Unix system calls, process management, and file handling, making it ideal for roles involving low-level programming, operating systems, or systems programming.