**EXPERIMENT 15**

**Design a C program to organize the file using two level directory structure.**

## AIM :

To design a C program to organize the file using two level directory structure

## Algorithm :

* 1. Define Structures: Define structures for files and directories. Each directory structure should contain an array for files and an array for subdirectories.
  2. Initialize Root Directory: Create a root directory structure. This serves as the starting point for the two-level directory structure.
  3. Add Files to Directories: Implement a function to add files to a specific directory. Handle adding files, updating the file count, and handling errors if the directory is full.
  4. Add Subdirectories: Implement a function to add subdirectories to a specific directory. Manage adding directories, updating the directory count, and handling errors if the parent directory is full.
  5. List Files and Subdirectories: Create functions to list all the files and subdirectories in a directory. These functions should iterate through the file and subdirectory arrays and print their names.
  6. Delete Files and Subdirectories (Optional): Implement functions to delete files and subdirectories from a directory. Handle removing files or directories, updating the counts, and handling errors if the file or directory is not found.
  7. Implement User Interface: Design a user interface for interacting with the program. This could be a menu-driven interface where users can add files, add subdirectories, list files, list subdirectories, delete files, delete subdirectories, or exit the program.
  8. Test the Program: Compile the program and test it thoroughly. Add files, add subdirectories, list files, list subdirectories, delete files, and delete subdirectories. Ensure the program handles different scenarios and errors gracefully.
  9. Refine and Expand (Optional): Refine the program based on testing results. Expand the functionality by adding more features, error handling, or optimizing the code.
  10. Document Your Code (Optional): Document your code by adding comments to explain the functionality of different sections. This will make it easier for others to understand the code in the future.

## A screenshot of a computer Description automatically generatedOUTPUT :