c) Hierarchical

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NAME_LEN 20
#define MAX_CHILDREN 10

// Define structure for the directory
struct Directory {
   char dirName[NAME_LEN];
   struct Directory* subDirs[MAX_CHILDREN];
   char* files[MAX_CHILDREN];
   int fileCount;
   int subDirCount;
};

// Function to create a new directory
struct Directory* createDirectory(char* name) {
   struct Directory* newDir = (struct Directory*)malloc(sizeof(struct Directory));
   strcpy(newDir->dirName, name);
   newDir->fileCount = 0;
   newDir->subDirCount = 0;
   return newDir;
}

// Function to create a file in a directory
void createFile(struct Directory* dir, char* fileName) {
   if (dir->fileCount < MAX_CHILDREN) {
      dir->files[dir->fileCount] = (char*)malloc(NAME_LEN * sizeof(char));
      strcpy(dir->files[dir->fileCount], fileName);
      dir->fileCount++;
      printf("File '%s' created in directory '%s'.\n", fileName, dir->dirName);
   } else {
      printf("Directory '%s' is full. Cannot create more files.\n", dir->dirName);
   }
}

// Function to create a subdirectory in a directory
void createSubDirectory(struct Directory* parentDir, char* subDirName) {
   if (parentDir->subDirCount < MAX_CHILDREN) {
      parentDir->subDirs[parentDir->subDirCount] = createDirectory(subDirName);
      parentDir->subDirCount++;
      printf("Subdirectory '%s' created under directory '%s'.\n", subDirName, parentDir-
>dirName);
   } else {
```

```
        printf("Directory '%s' has reached the maximum subdirectory limit.\n", parentDir->dirName);
    }
}

// Function to display files and subdirectories
void displayDirectory(struct Directory* dir, int level) {
    for (int i = 0; i < level; i++)
        printf("  ");
    printf("Directory: %s\n", dir->dirName);

    for (int i = 0; i < dir->fileCount; i++) {
        for (int j = 0; j < level + 1; j++)
            printf("  ");
        printf("File: %s\n", dir->files[i]);
    }

    for (int i = 0; i < dir->subDirCount; i++) {
        displayDirectory(dir->subDirs[i], level + 1);
    }
}

// Main function to simulate the hierarchical directory structure
int main() {
    struct Directory* root = createDirectory("Root");
    int choice;
    char dirName[NAME_LEN], fileName[NAME_LEN];
    struct Directory* currentDir = root;

    while (1) {
        printf("\nHierarchical Directory File System Simulation\n");
        printf("1. Create File\n2. Create Subdirectory\n3. Display Directory\n4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter file name to create: ");
                scanf("%s", fileName);
                createFile(currentDir, fileName);
                break;

            case 2:
                printf("Enter subdirectory name to create: ");
                scanf("%s", dirName);
                createSubDirectory(currentDir, dirName);
                break;
```

```
        case 3:
            displayDirectory(root, 0);
            break;

        case 4:
            printf("Exiting...\n");
            return 0;

        default:
            printf("Invalid choice.\n");
    }
  }

  return 0;
}
```

```
Hierarchical Directory File System Simulation
1. Create File
2. Create Subdirectory
3. Display Directory
4. Exit
Enter your choice: 1
Enter file name to create: shreyas
File 'shreyas' created in directory 'Root'.

Hierarchical Directory File System Simulation
1. Create File
2. Create Subdirectory
3. Display Directory
4. Exit
Enter your choice: 2
Enter subdirectory name to create: suhas
Subdirectory 'suhas' created under directory 'Root'.
```

```
Hierarchical Directory File System Simulation
1. Create File
2. Create Subdirectory
3. Display Directory
4. Exit
Enter your choice: 3
Directory: Root
  File: shreyas
  Directory: suhas

Hierarchical Directory File System Simulation
1. Create File
2. Create Subdirectory
3. Display Directory
4. Exit
Enter your choice: 4
Exiting...


=== Code Execution Successful ===
```