```
// Jessica Elkins
// CS332
// Lab 5
// 2/20/20
// This program takes pathway name as a command line argument and traverses the
// file hierarchy recursively and lists all the sub directories and the files in
// the sub directories.


#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>


void fileTraversal(char *name) {

        //initializing count at one
        int count = 1;

        //to store DIR pointer returned from opendir
        DIR *dir;
        // to store pointer to structure returned from readdir
        struct dirent *dirent;

        //opening directory
        dir = opendir(name);

        // if not able to open directory
        if(dir == NULL) {
                // print error message and terminate program
                printf("Error while opening directory. Exiting. \n");
                exit(-1);
        }

        // readdir returns NULL at end or directory or error
        while((dirent = readdir(dir)) != NULL) {
                //if path name is directory
                if((dirent->d_type == DT_DIR) && (strcmp(dirent->d_name, ".") != 0) &&
(strcmp(dirent->d_name, "..") != 0)) {
                        //allocating size for pathName
                        char pathName[BUFSIZ];
                        //using snprintf to format pathway name and storing it in pathN
ame
                        snprintf(pathName, sizeof(pathName), "%s/%s", name, dirent->d_n
ame);
                        printf(" \n");
                        //displaying directory name
                        printf("%d. DIR: %s \n", count, dirent->d_name);
                        //recursively call function to traverse directory
                        fileTraversal(pathName);
                }else{
                        //if not directory, just list file name
                        printf("%d. %s \n", count, dirent->d_name);
                }

                //increase count
                count++;
        }

        printf(" \n");

        //close directory
        closedir(dir);
```

```c
}

int main(int argc, char **argv) {

        //if only one command line argument was given
        if(argc < 2) {
                printf("Usage: %s <directoryname> \nExiting. \n", argv[0]);
                exit(-1);
        }

        //calling fileTraverse with 2nd element in argv as parameter
        fileTraversal(argv[1]);

        return 0;
}
```