```
/* Sample C program to illustrate the use of opendir, readdir, and closedir
 * system calls.
 * This version prints the file type next to the file name.
 */
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>
char *filetype(unsigned char type)
{
    char *str;
    switch (type)
    {
    case DT_BLK:
        str = "block device";
        break;
    case DT_CHR:
        str = "character device";
        break;
    case DT_DIR:
        str = "directory";
        break;
    case DT_FIFO:
        str = "named pipe (FIFO)";
        break;
    case DT_LNK:
        str = "symbolic link";
        break;
    case DT_REG:
        str = "regular file";
        break;
    case DT_SOCK:
        str = "UNIX domain socket";
        break;
    case DT_UNKNOWN:
        str = "unknown file type";
        break;
    default:
        str = "UNKNOWN";
    }
    return str;
}

int recursive(DIR *parentDir, char *path)
{
    struct dirent *dirent;
    int count = 1;
    while ((dirent = readdir(parentDir)) != NULL)
    {
        if (dirent->d_type == DT_DIR)
        {
            if (strcmp(dirent->d_name, ".") == 0 || (strcmp(dirent->d_name, "..") == 0)
)
            {
                continue;
            }
            path = strcat(path, "/");
            path = strcat(path, dirent->d_name);
            parentDir = opendir(path);
            if (parentDir == NULL)
            {
                printf("Error opening directory ’%s’\n", dirent->d_name);
                exit(-1);
            }
            recursive(parentDir, path);
        }
        printf("  [%d] %s (%s)\n", count, dirent->d_name, filetype(dirent->d_type));
        count++;
    }
    closedir(parentDir);
    return 0;
}

int main(int argc, char **argv)
{
    DIR *parentDir;
    if (argc < 2)
    {
        printf("Usage: %s <dirname>\n", argv[0]);
        exit(-1);
    }
    parentDir = opendir(argv[1]);
    if (parentDir == NULL)
    {
        printf("Error opening directory ’%s’\n", argv[1]);
        exit(-1);
    }
    recursive(parentDir, argv[1]);
    return 0;
}
```