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

#define MAX_FILES 10
#define MAX_SUBDIRS 5
#define MAX_FILE_NAME 20
#define MAX_DIR_NAME 20

typedef struct {
  char name[MAX_FILE_NAME];
} File;

typedef struct {
  char name[MAX_DIR_NAME];
  File files[MAX_FILES];
  int num_files;
} Directory;

typedef struct {
  char name[MAX_DIR_NAME];
  Directory subdirs[MAX_SUBDIRS];
  int num_subdirs;
} TwoLevelDirectory;

typedef struct HierarchicalDirectory HierarchicalDirectory;

struct HierarchicalDirectory {
  char name[MAX_DIR_NAME];
  HierarchicalDirectory *subdirs[MAX_SUBDIRS];
  int num_subdirs;
  File files[MAX_FILES];
  int num_files;
};

void initialize_directory(Directory *dir) { dir->num_files = 0; }

void add_file(Directory *dir, const char *file_name) {
  if (dir->num_files < MAX_FILES) {
    strcpy(dir->files[dir->num_files].name, file_name);
    dir->num_files++;
  } else {
    printf("Directory is full.\n");
  }
}

void print_single_level_directory(Directory *dir) {
  printf("Files in directory '%s':\n", dir->name);
  for (int i = 0; i < dir->num_files; i++) {
    printf("  %s\n", dir->files[i].name);
  }
}

void initialize_two_level_directory(TwoLevelDirectory *two_level_dir) {
  two_level_dir->num_subdirs = 0;
}

void add_subdir(TwoLevelDirectory *two_level_dir, const char *subdir_name) {
  if (two_level_dir->num_subdirs < MAX_SUBDIRS) {
    strcpy(two_level_dir->subdirs[two_level_dir->num_subdirs].name,
           subdir_name);
    initialize_directory(&two_level_dir->subdirs[two_level_dir->num_subdirs]);
    two_level_dir->num_subdirs++;
  } else {
```

```
    printf("Two-level directory is full.\n");
  }
}

void print_two_level_directory(TwoLevelDirectory *two_level_dir) {
  printf("Two-Level Directory '%s':\n", two_level_dir->name);
  for (int i = 0; i < two_level_dir->num_subdirs; i++) {
    printf("  Subdirectory '%s':\n", two_level_dir->subdirs[i].name);
    for (int j = 0; j < two_level_dir->subdirs[i].num_files; j++) {
      printf("    %s\n", two_level_dir->subdirs[i].files[j].name);
    }
  }
}

void initialize_hierarchical_directory(HierarchicalDirectory *hier_dir,
                                       const char *name) {
  strcpy(hier_dir->name, name);
  hier_dir->num_files = 0;
  hier_dir->num_subdirs = 0;
}

void add_file_hierarchical(HierarchicalDirectory *hier_dir,
                           const char *file_name) {
  if (hier_dir->num_files < MAX_FILES) {
    strcpy(hier_dir->files[hier_dir->num_files].name, file_name);
    hier_dir->num_files++;
  } else {
    printf("Hierarchical directory is full.\n");
  }
}

void add_subdir_hierarchical(HierarchicalDirectory *hier_dir,
                             HierarchicalDirectory *subdir) {
  if (hier_dir->num_subdirs < MAX_SUBDIRS) {
    hier_dir->subdirs[hier_dir->num_subdirs] = subdir;
    hier_dir->num_subdirs++;
  } else {
    printf("Hierarchical directory has reached maximum subdirectories.\n");
  }
}

void print_hierarchical_directory(HierarchicalDirectory *hier_dir, int level) {
  for (int i = 0; i < level; i++) {
    printf("  ");
  }
  printf("Directory '%s':\n", hier_dir->name);
  for (int i = 0; i < hier_dir->num_files; i++) {
    for (int j = 0; j < level + 1; j++) {
      printf("  ");
    }
    printf("File: %s\n", hier_dir->files[i].name);
  }
  for (int i = 0; i < hier_dir->num_subdirs; i++) {
    print_hierarchical_directory(hier_dir->subdirs[i], level + 1);
  }
}

int main() {
  Directory single_level_dir;
  strcpy(single_level_dir.name, "Root");
  initialize_directory(&single_level_dir);
  add_file(&single_level_dir, "file1.txt");
  add_file(&single_level_dir, "file2.txt");
  print_single_level_directory(&single_level_dir);
```

```
    TwoLevelDirectory two_level_dir;
    strcpy(two_level_dir.name, "MainDir");
    initialize_two_level_directory(&two_level_dir);
    add_subdir(&two_level_dir, "SubDir1");
    add_file(&two_level_dir.subdirs[0], "file3.txt");
    add_file(&two_level_dir.subdirs[0], "file4.txt");
    add_subdir(&two_level_dir, "SubDir2");
    add_file(&two_level_dir.subdirs[1], "file5.txt");
    print_two_level_directory(&two_level_dir);

    HierarchicalDirectory root_dir;
    initialize_hierarchical_directory(&root_dir, "RootHier");
    HierarchicalDirectory subdir1;
    initialize_hierarchical_directory(&subdir1, "SubDirHier1");
    HierarchicalDirectory subdir2;
    initialize_hierarchical_directory(&subdir2, "SubDirHier2");

    add_file_hierarchical(&root_dir, "file6.txt");
    add_file_hierarchical(&root_dir, "file7.txt");
    add_file_hierarchical(&subdir1, "file8.txt");
    add_file_hierarchical(&subdir2, "file9.txt");

    add_subdir_hierarchical(&root_dir, &subdir1);
    add_subdir_hierarchical(&root_dir, &subdir2);

    print_hierarchical_directory(&root_dir, 0);

    return 0;
}
```