Exp – 12a)

AIM: To write a C code to implement Single level Directory file orgranisation structure

PROGRAM:

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

#define MAX_FILES 100
#define MAX_FILENAME_LEN 100

struct Directory {
    char filename[MAX_FILES][MAX_FILENAME_LEN];
    int filecount;
};

void createFile(struct Directory *dir);
void deleteFile(struct Directory *dir);
void displayFiles(const struct Directory *dir);

int main() {
    struct Directory dir;
    dir.filecount = 0;

    int choice;

    while (1) {
        printf("\n--- File Management Menu ---\n");
        printf("1. Create File\n");
        printf("2. Delete File\n");
        printf("3. Display Files\n");
        printf("4. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                createFile(&dir);
                break;
            case 2:
                deleteFile(&dir);
                break;
            case 3:
                displayFiles(&dir);
                break;
            case 4:
                return 0;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
}
```

```c
void createFile(struct Directory *dir) {
    if (dir->filecount >= MAX_FILES) {
        printf("Directory is full. Cannot create more files.\n");
        return;
    }

    char name[MAX_FILENAME_LEN];
    printf("Enter file name: ");
    scanf("%s", name);
    for (int i = 0; i < dir->filecount; i++) {
        if (strcmp(name, dir->filename[i]) == 0) {
            printf("File already exists!\n");
            return;
        }
    }

    strcpy(dir->filename[dir->filecount++], name);
    printf("File created.\n");
}

void deleteFile(struct Directory *dir) {
    char name[MAX_FILENAME_LEN];
    printf("Enter file name to delete: ");
    scanf("%s", name);

    for (int i = 0; i < dir->filecount; i++) {
        if (strcmp(name, dir->filename[i]) == 0) {
            strcpy(dir->filename[i], dir->filename[--dir->filecount]);
            printf("File deleted.\n");
            return;
        }
    }

    printf("File not found!\n");
}

void displayFiles(const struct Directory *dir) {
    if (dir->filecount == 0) {
        printf("No files found.\n");
        return;
    }

    printf("Files:\n");
    for (int i = 0; i < dir->filecount; i++) {
        printf("%s\n", dir->filename[i]);
    }
}
```

OUTPUT:

```
jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ ./singleLevel

--- File Management Menu ---
1. Create File
2. Delete File
3. Display Files
4. Exit
Enter choice: 1
Enter file name: sample1.txt
File created.

--- File Management Menu ---
1. Create File
2. Delete File
3. Display Files
4. Exit
Enter choice: 1
Enter file name: sample2.txt
File created.

--- File Management Menu ---
1. Create File
2. Delete File
3. Display Files
4. Exit
Enter choice: 3
Files:
sample1.txt
sample2.txt

--- File Management Menu ---
1. Create File
2. Delete File
3. Display Files
4. Exit
Enter choice: 4
```

Exp – 12b)

AIM: To write a C code to implement Two-level directory file orgranisation structure

PROGRAM:

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

#define MAX_USERS 10
#define MAX_FILES 10
#define NAME_LEN 100

struct File {
    char name[NAME_LEN];
};

struct User {
    char name[NAME_LEN];
    struct File files[MAX_FILES];
    int filecount;
};

void createUser(struct User users[], int *usercount);
void createFile(struct User users[], int usercount);
void displayDirectory(const struct User users[], int usercount);

int main() {
    struct User users[MAX_USERS];
    int usercount = 0;
    int choice;
    while (1) {
        printf("\n--- Directory Management ---\n");
        printf("1. Create User\n");
        printf("2. Create File\n");
        printf("3. Display Directory\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                createUser(users, &usercount);
                break;
            case 2:
                createFile(users, usercount);
                break;
            case 3:
                displayDirectory(users, usercount);
                break;
            case 4:
                return 0;
            default:
                printf("Invalid choice. Try again.\n");
        }
```

```
    }
}

void createUser(struct User users[], int *usercount) {
    if (*usercount >= MAX_USERS) {
        printf("Maximum user limit reached.\n");
        return;
    }

    printf("Enter user name: ");
    scanf("%s", users[*usercount].name);
    users[*usercount].filecount = 0;
    (*usercount)++;
    printf("User created successfully.\n");
}

void createFile(struct User users[], int usercount) {
    char uname[NAME_LEN], fname[NAME_LEN];
    printf("Enter user name: ");
    scanf("%s", uname);

    for (int i = 0; i < usercount; i++) {
        if (strcmp(uname, users[i].name) == 0) {
            if (users[i].filecount >= MAX_FILES) {
                printf("User has reached max file limit.\n");
                return;
            }
            printf("Enter file name: ");
            scanf("%s", fname);

            for (int j = 0; j < users[i].filecount; j++) {
                if (strcmp(fname, users[i].files[j].name) == 0) {
                    printf("File already exists.\n");
                    return;
                }
            }

            strcpy(users[i].files[users[i].filecount++].name, fname);
            printf("File created successfully.\n");
            return;
        }
    }
    printf("User not found.\n");
}
```

```
void displayDirectory(const struct User users[], int usercount) {
    if (usercount == 0) {
        printf("No users available.\n");
        return;
    }

    for (int i = 0; i < usercount; i++) {
        printf("\nUser: %s\nFiles:\n", users[i].name);
        if (users[i].filecount == 0) {
            printf("  No files.\n");
        } else {
            for (int j = 0; j < users[i].filecount; j++) {
                printf("  %s\n", users[i].files[j].name);
            }
        }
    }
}
```

OUTPUT:

```
jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ ./twoLevel

--- Directory Management ---
1. Create User
2. Create File
3. Display Directory
4. Exit
Enter your choice: 1
Enter user name: Jagadesh
User created successfully.

--- Directory Management ---
1. Create User
2. Create File
3. Display Directory
4. Exit
Enter your choice: 2
Enter user name: Jagadesh
Enter file name: sample.txt
File created successfully.

--- Directory Management ---
1. Create User
2. Create File
3. Display Directory
4. Exit
Enter your choice: 3

User: Jagadesh
Files:
  sample.txt

--- Directory Management ---
1. Create User
2. Create File
3. Display Directory
4. Exit
Enter your choice: 4
```