

15. Design a C program to organize the file using two level directory structure.

```
#include <stdio.h>
#include <string.h>
struct {
    char dname[10];
    char fname[10][10];
    int fcount;
} dir[10];
int main() {
    int dc = 0, ch, i, j;
    char d[10], f[10];
    while (1) {
        printf("\n1. Create Directory\n2. Create File\n3. Delete File\n4. Search File\n5.
Display\n6. Exit\nEnter choice: ");
        scanf("%d", &ch);
        switch (ch) {
            case 1:
                printf("Enter directory name: ");
                scanf("%s", dir[dc].dname);
                dir[dc].fcount = 0;
                dc++;
                break;
            case 2:
                printf("Enter directory name: ");
                scanf("%s", d);
                for (i = 0; i < dc; i++) {
                    if (strcmp(d, dir[i].dname) == 0) {
                        printf("Enter file name: ");
                        scanf("%s", dir[i].fname[dir[i].fcount]);
                        dir[i].fcount++;
                    }
                }
            case 3:
            case 4:
            case 5:
            case 6:
                break;
        }
    }
}
```

```
        break;
    }
}
```

```
break;
```

case 3:

```
printf("Enter directory name: ");
scanf("%s", d);
printf("Enter file name to delete: ");
scanf("%s", f);
for (i = 0; i < dc; i++) {
    if (strcmp(d, dir[i].dname) == 0) {
        for (j = 0; j < dir[i].fcount; j++) {
            if (strcmp(f, dir[i].fname[j]) == 0) {
                strcpy(dir[i].fname[j], dir[i].fname[dir[i].fcount - 1]);
                dir[i].fcount--;
                printf("File deleted.\n");
                goto next;
            }
        }
    }
}
printf("File not found.\n");
```

next:

```
break;
```

case 4:

```
printf("Enter directory name: ");
scanf("%s", d);
printf("Enter file name to search: ");
scanf("%s", f);
for (i = 0; i < dc; i++) {
```

```

        if (strcmp(d, dir[i].dname) == 0) {
            for (j = 0; j < dir[i].fcount; j++) {
                if (strcmp(f, dir[i].fname[j]) == 0) {
                    printf("File found.\n");
                    goto next2;
                }
            }
        }
        printf("File not found.\n");
next2:
        break;
case 5:
        for (i = 0; i < dc; i++) {
            printf("\nDirectory %s contains:\n", dir[i].dname);
            for (j = 0; j < dir[i].fcount; j++)
                printf("%s\n", dir[i].fname[j]);
        }
        break;
case 6:
        return 0;
    }
}
}

```

OUTPUT:

1. Create Directory
2. Create File
3. Delete File

4. Search File

5. Display

6. Exit

Enter choice: 1

Enter directory name: OS

1. Create Directory

2. Create File

3. Delete File

4. Search File

5. Display

6. Exit

Enter choice: 2

Enter directory name: RAM

1. Create Directory

2. Create File

3. Delete File

4. Search File

5. Display

6. Exit

Enter choice: 3

Enter directory name: PROCESSOR

Enter file name to delete: OS

File not found.

1. Create Directory

2. Create File

3. Delete File

4. Search File

5. Display

6. Exit

Enter choice: 4

Enter directory name: VD

Enter file name to search: OS

File not found.

1. Create Directory

2. Create File

3. Delete File

4. Search File

5. Display

6. Exit

Enter choice: 5

Directory OS contains:

1. Create Directory

2. Create File

3. Delete File

4. Search File

5. Display

6. Exit

Enter choice: 6