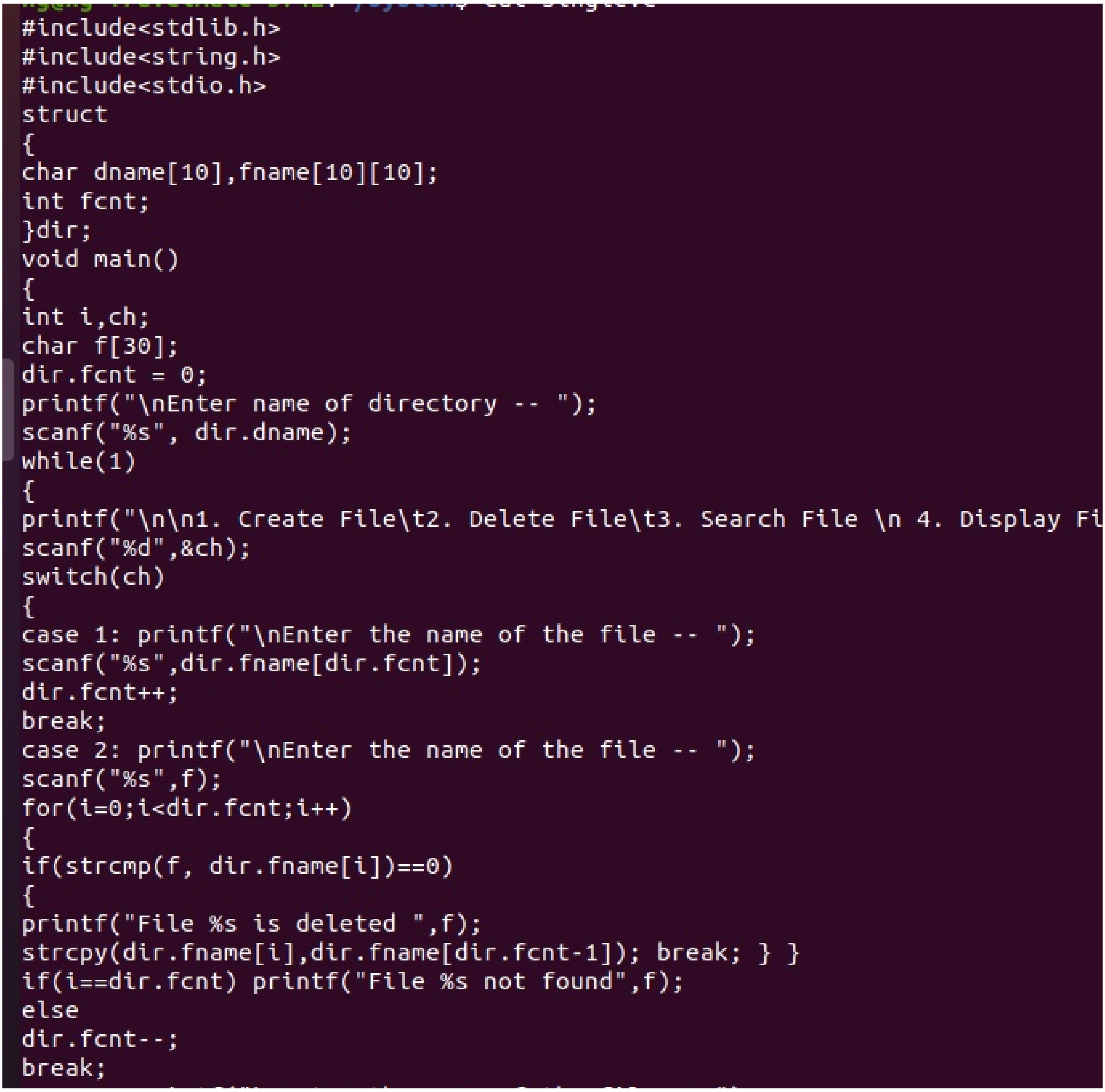
PROGRAM NUMBER :4

AIM:

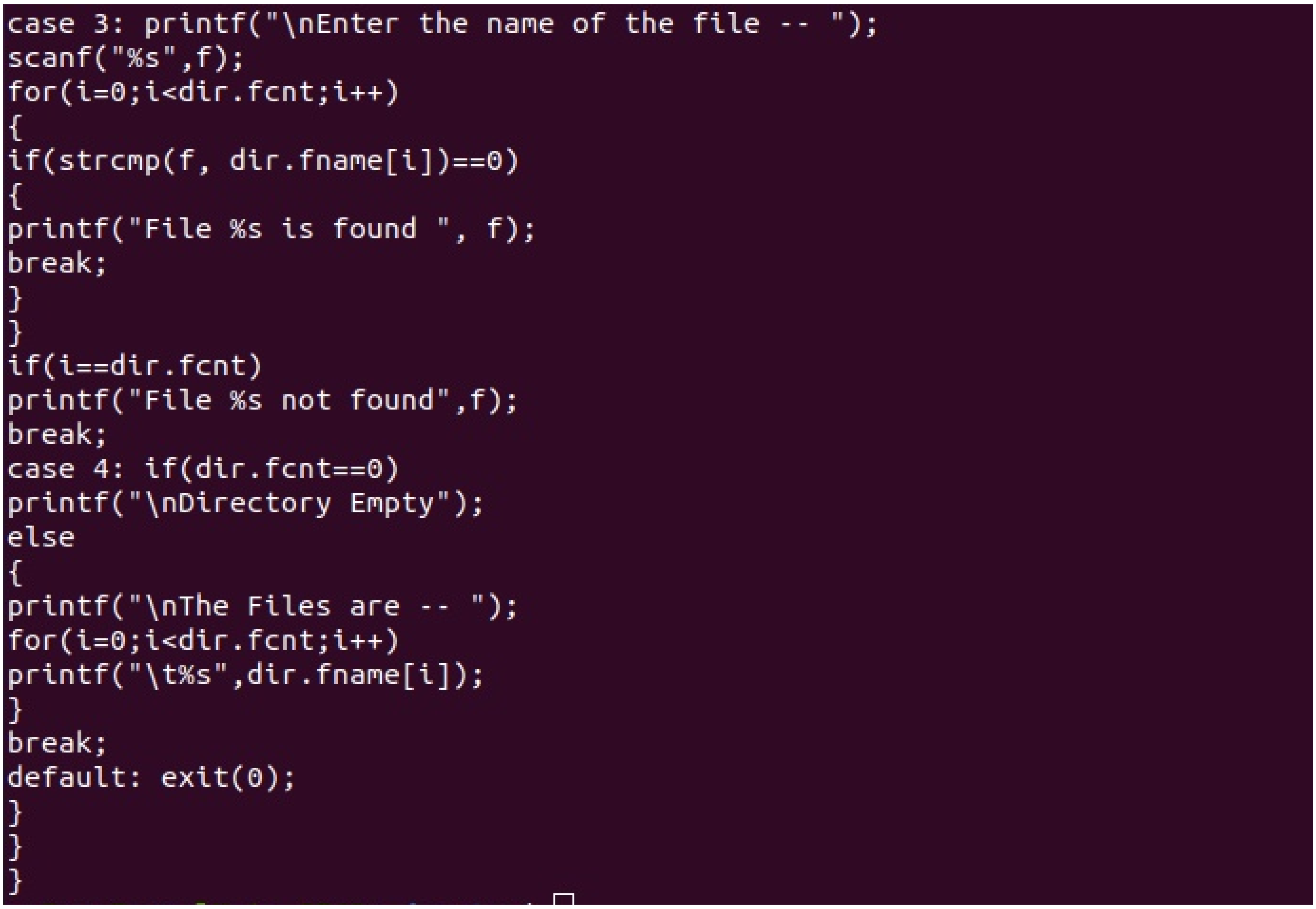
Simulate the following file organization techniques
a) Single level directory
b) Two level directory
c) Hierarchical

PROGRAM

a) Single level directory

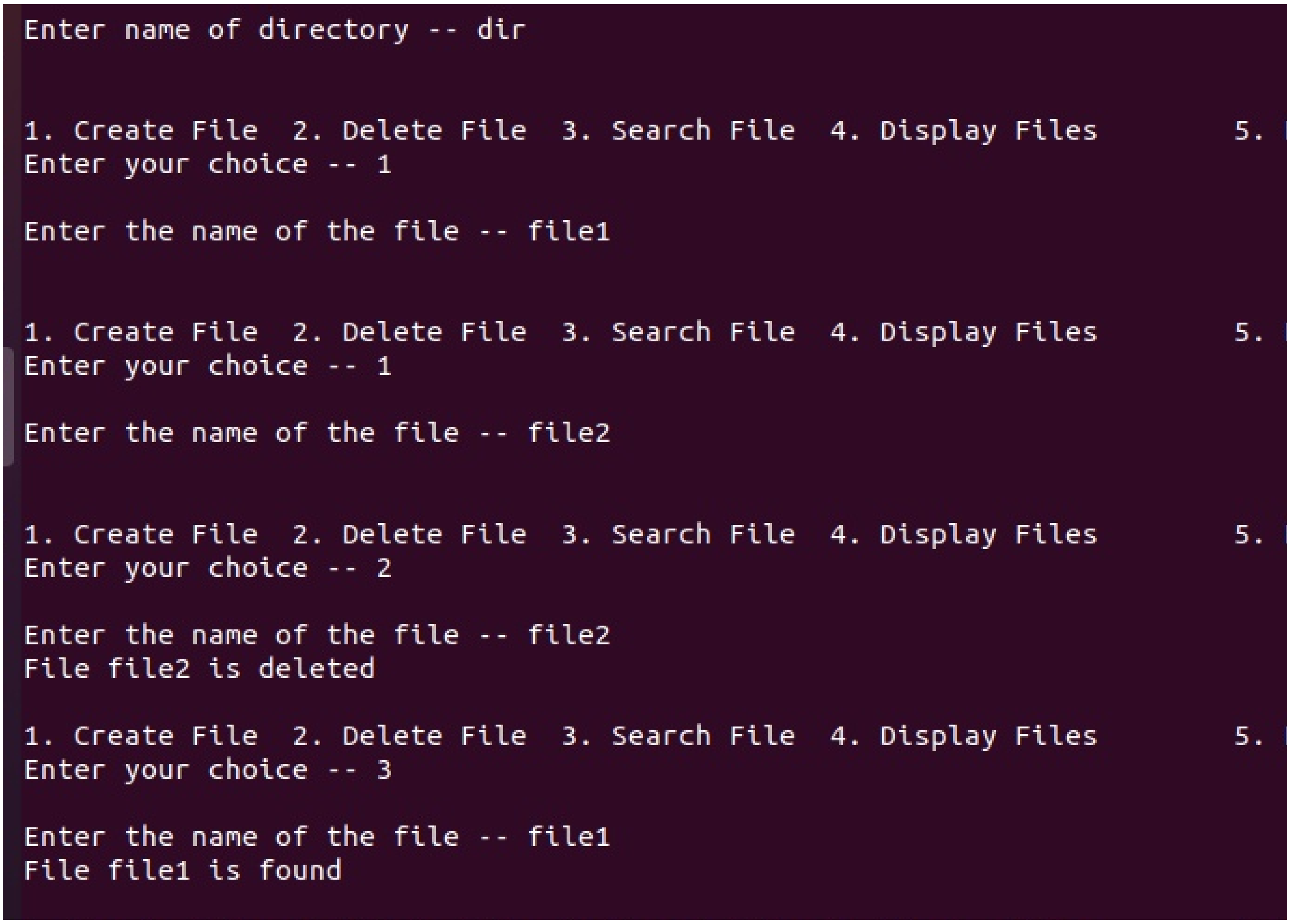


```
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
struct
{
char dname[10],fname[10][10];
int fcnt;
}dir;
void main()
{
int i,ch;
char f[30];
dir.fcnt = 0;
printf("\nEnter name of directory -- ");
scanf("%s", dir.dname);
while(1)
{
printf("\n\n1. Create File\t2. Delete File\t3. Search File \n 4. Display Fi
scanf("%d",&ch);
switch(ch)
{
case 1: printf("\nEnter the name of the file -- ");
scanf("%s",dir.fname[dir.fcnt]);
dir.fcnt++;
break;
case 2: printf("\nEnter the name of the file -- ");
scanf("%s",f);
for(i=0;i<dir.fcnt;i++)
{
if(strcmp(f, dir.fname[i])==0)
{
printf("File %s is deleted ",f);
strcpy(dir.fname[i],dir.fname[dir.fcnt-1]); break; } }
if(i==dir.fcnt) printf("File %s not found",f);
else
dir.fcnt--;
break;
```



```
case 3: printf("\nEnter the name of the file -- ");
scanf("%s",f);
for(i=0;i<dir.fcnt;i++)
{
if(strcmp(f, dir.fname[i])==0)
{
printf("File %s is found ", f);
break;
}
}
if(i==dir.fcnt)
printf("File %s not found",f);
break;
case 4: if(dir.fcnt==0)
printf("\nDirectory Empty");
else
{
printf("\nThe Files are -- ");
for(i=0;i<dir.fcnt;i++)
printf("\t%s",dir.fname[i]);
}
break;
default: exit(0);
}
}
}
```

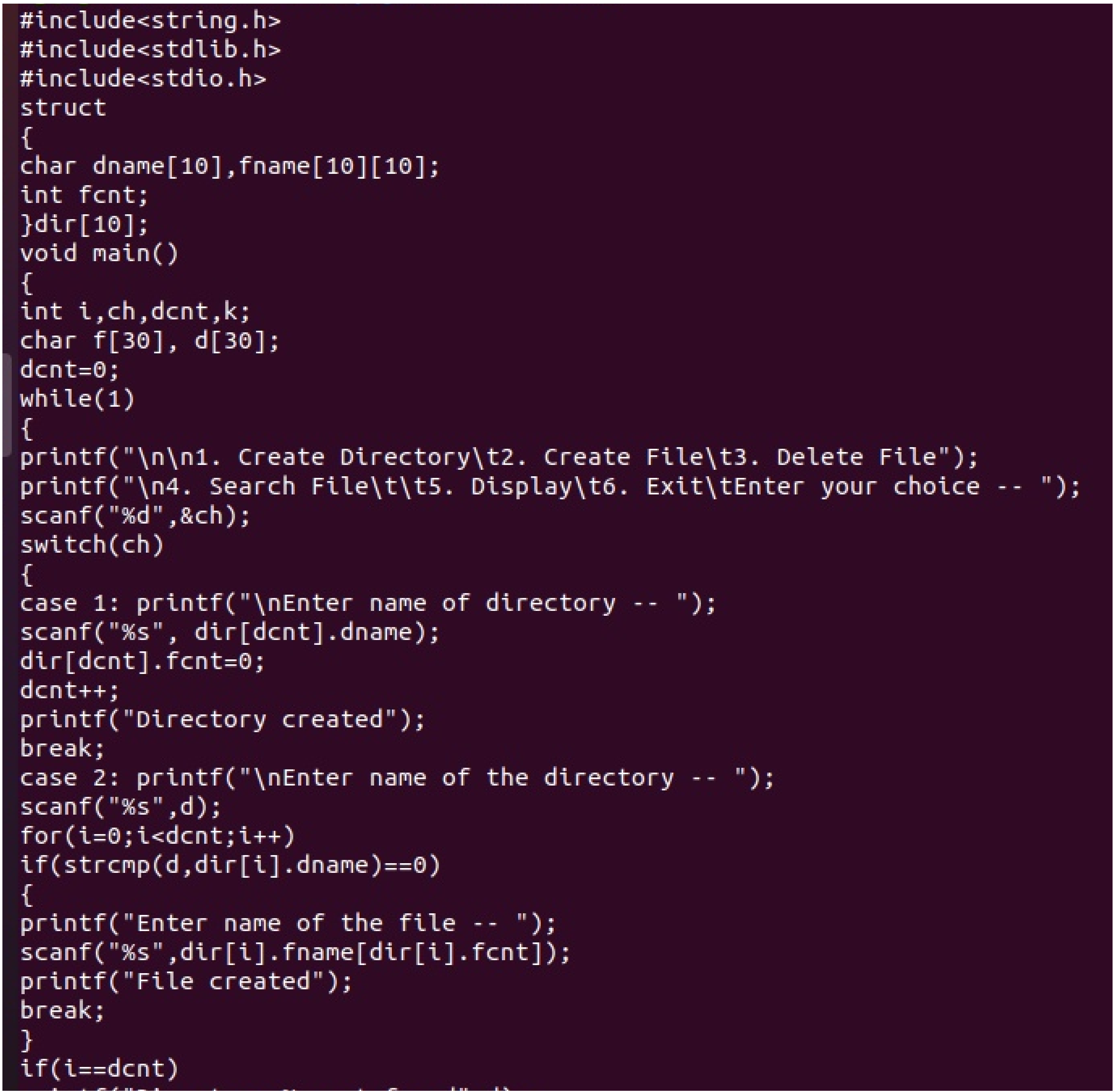
OUTPUT



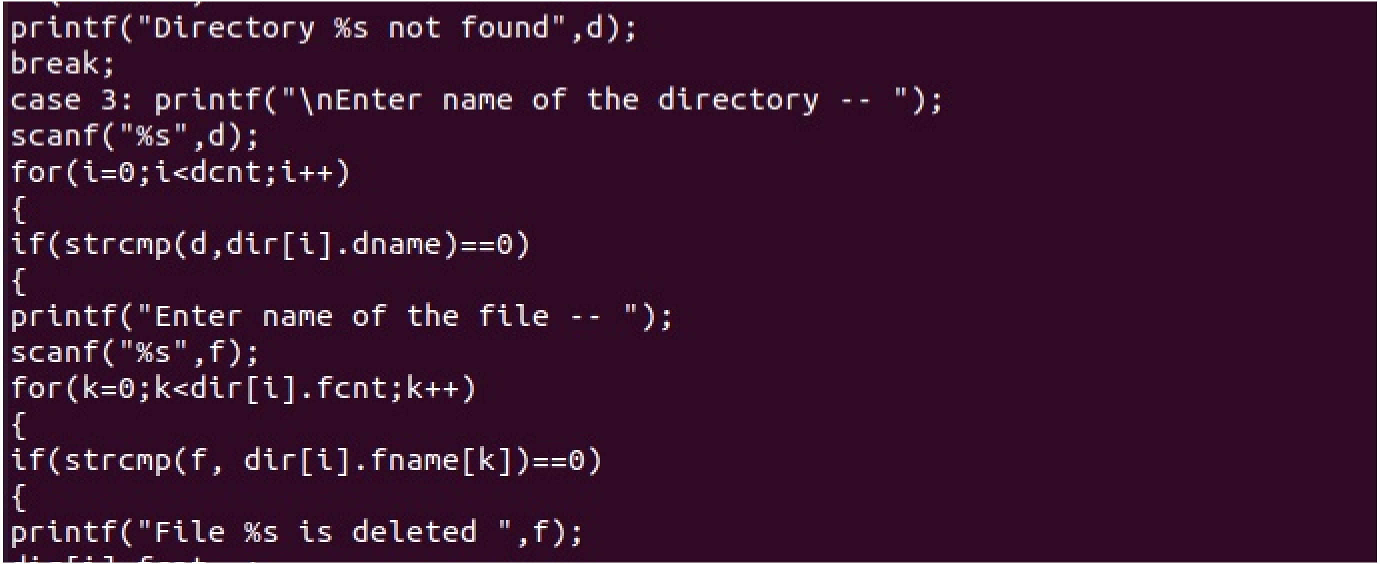
```
Enter name of directory -- dir


1. Create File  2. Delete File  3. Search File  4. Display Files       5.
Enter your choice -- 1

Enter the name of the file -- file1


1. Create File  2. Delete File  3. Search File  4. Display Files       5.
Enter your choice -- 1

Enter the name of the file -- file2


1. Create File  2. Delete File  3. Search File  4. Display Files       5.
Enter your choice -- 2

Enter the name of the file -- file2
File file2 is deleted

1. Create File  2. Delete File  3. Search File  4. Display Files       5.
Enter your choice -- 3

Enter the name of the file -- file1
File file1 is found
```

PROGRAM

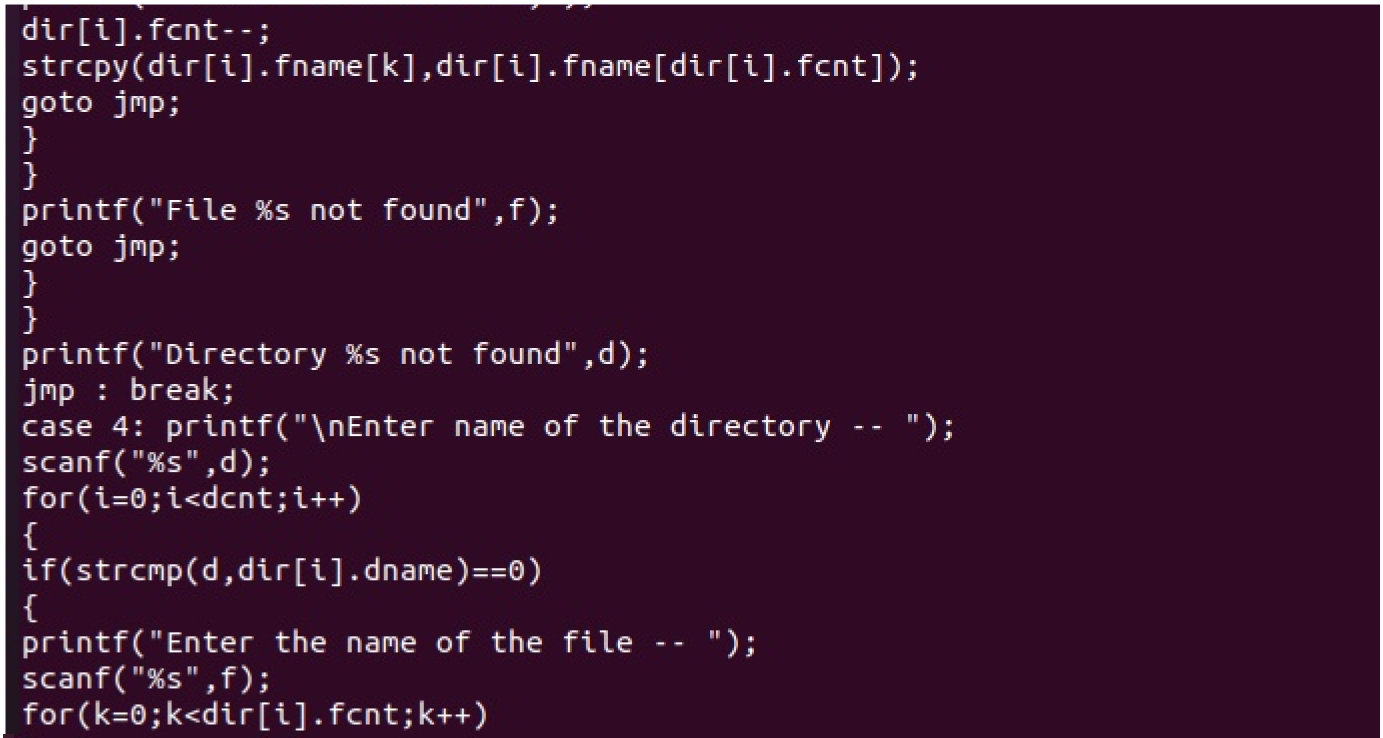
b) Two level directory



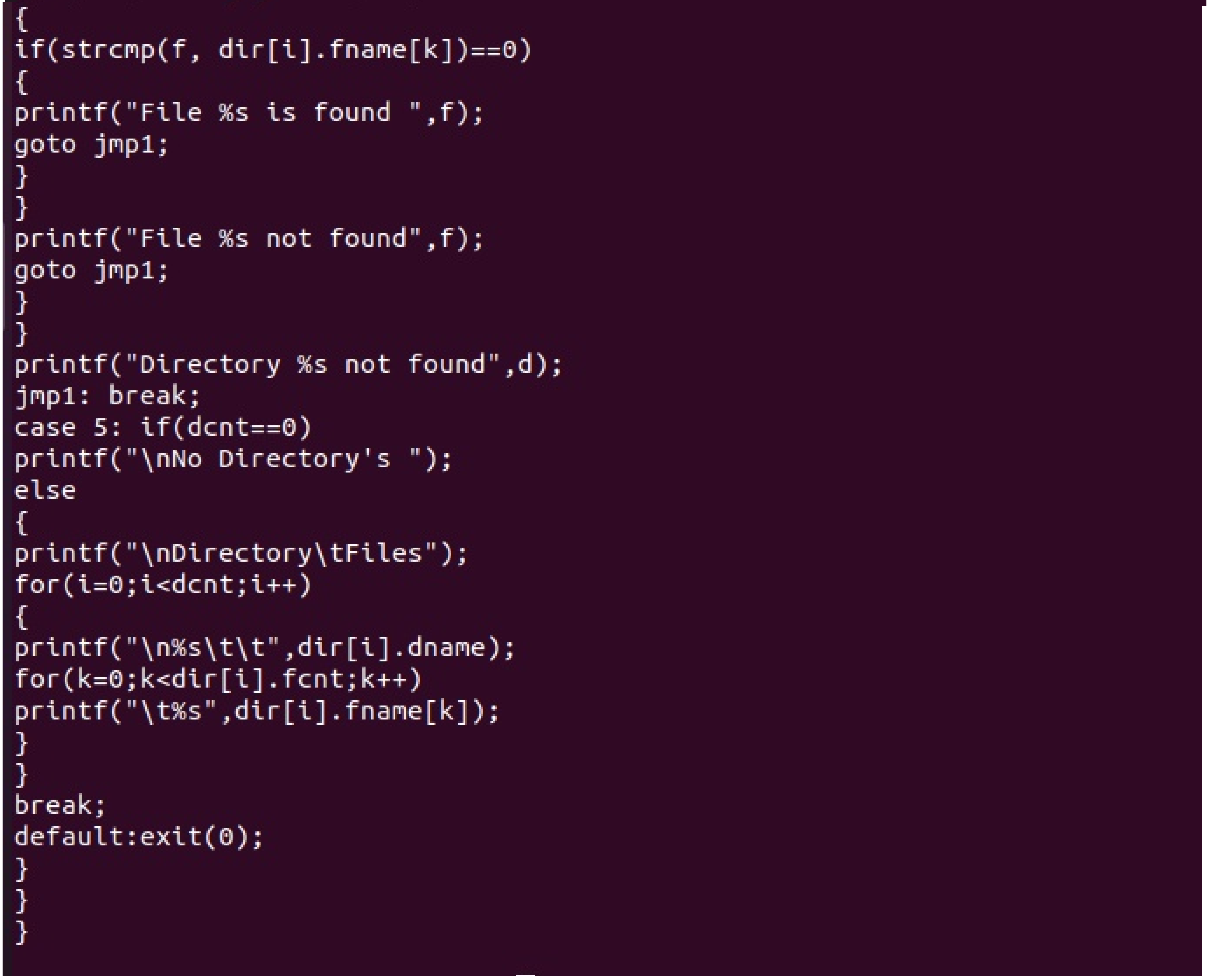
```
#include<string.h>
#include<stdlib.h>
#include<stdio.h>
struct
{
char dname[10],fname[10][10];
int fcnt;
}dir[10];
void main()
{
int i,ch,dcnt,k;
char f[30], d[30];
dcnt=0;
while(1)
{
printf("\n\n1. Create Directory\t2. Create File\t3. Delete File");
printf("\n4. Search File\t\t5. Display\t6. Exit\tEnter your choice -- ");
scanf("%d",&ch);
switch(ch)
{
case 1: printf("\nEnter name of directory -- ");
scanf("%s", dir[dcnt].dname);
dir[dcnt].fcnt=0;
dcnt++;
printf("Directory created");
break;
case 2: printf("\nEnter name of the directory -- ");
scanf("%s",d);
for(i=0;i<dcnt;i++)
if(strcmp(d,dir[i].dname)==0)
{
printf("Enter name of the file -- ");
scanf("%s",dir[i].fname[dir[i].fcnt]);
printf("File created");
break;
}
if(i==dcnt)
```



```
printf("Directory %s not found",d);
break;
case 3: printf("\nEnter name of the directory -- ");
scanf("%s",d);
for(i=0;i<dcnt;i++)
{
if(strcmp(d,dir[i].dname)==0)
{
printf("Enter name of the file -- ");
scanf("%s",f);
for(k=0;k<dir[i].fcnt;k++)
{
if(strcmp(f, dir[i].fname[k])==0)
{
printf("File %s is deleted ",f);
```

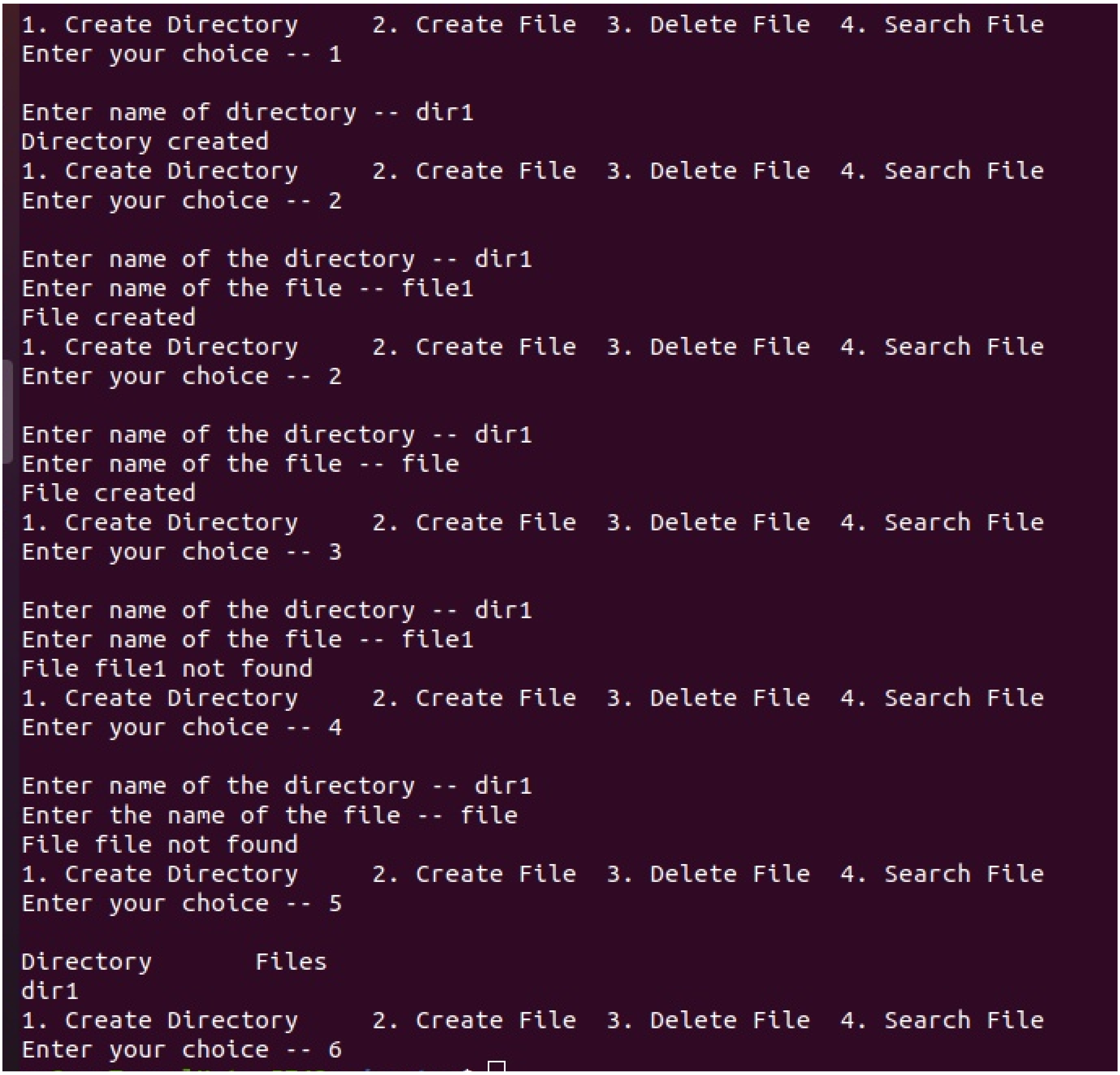


```
dir[i].fcnt--;
strcpy(dir[i].fname[k],dir[i].fname[dir[i].fcnt]);
goto jmp;
}
}
printf("File %s not found",f);
goto jmp;
}
}
printf("Directory %s not found",d);
jmp : break;
case 4: printf("\nEnter name of the directory -- ");
scanf("%s",d);
for(i=0;i<dcnt;i++)
{
if(strcmp(d,dir[i].dname)==0)
{
printf("Enter the name of the file -- ");
scanf("%s",f);
for(k=0;k<dir[i].fcnt;k++)
```



```
{
if(strcmp(f, dir[i].fname[k])==0)
{
printf("File %s is found ",f);
goto jmp1;
}
}
printf("File %s not found",f);
goto jmp1;
}
}
printf("Directory %s not found",d);
jmp1: break;
case 5: if(dcnt==0)
printf("\nNo Directory's ");
else
{
printf("\nDirectory\tFiles");
for(i=0;i<dcnt;i++)
{
printf("\n%s\t\t",dir[i].dname);
for(k=0;k<dir[i].fcnt;k++)
printf("\t%s",dir[i].fname[k]);
}
}
break;
default:exit(0);
}
}
}
```

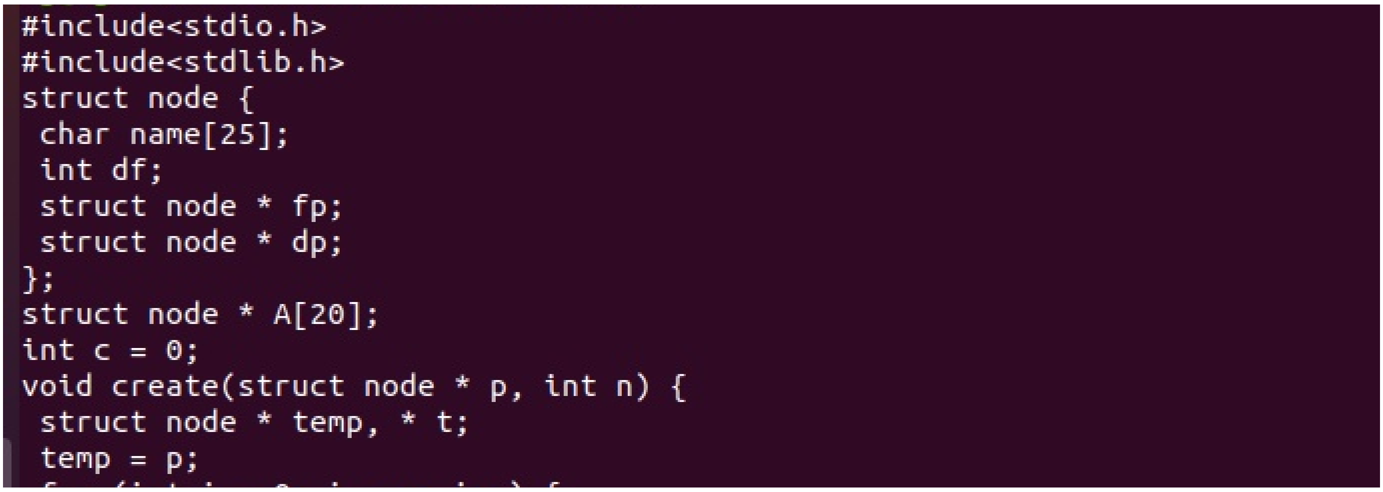
OUTPUT



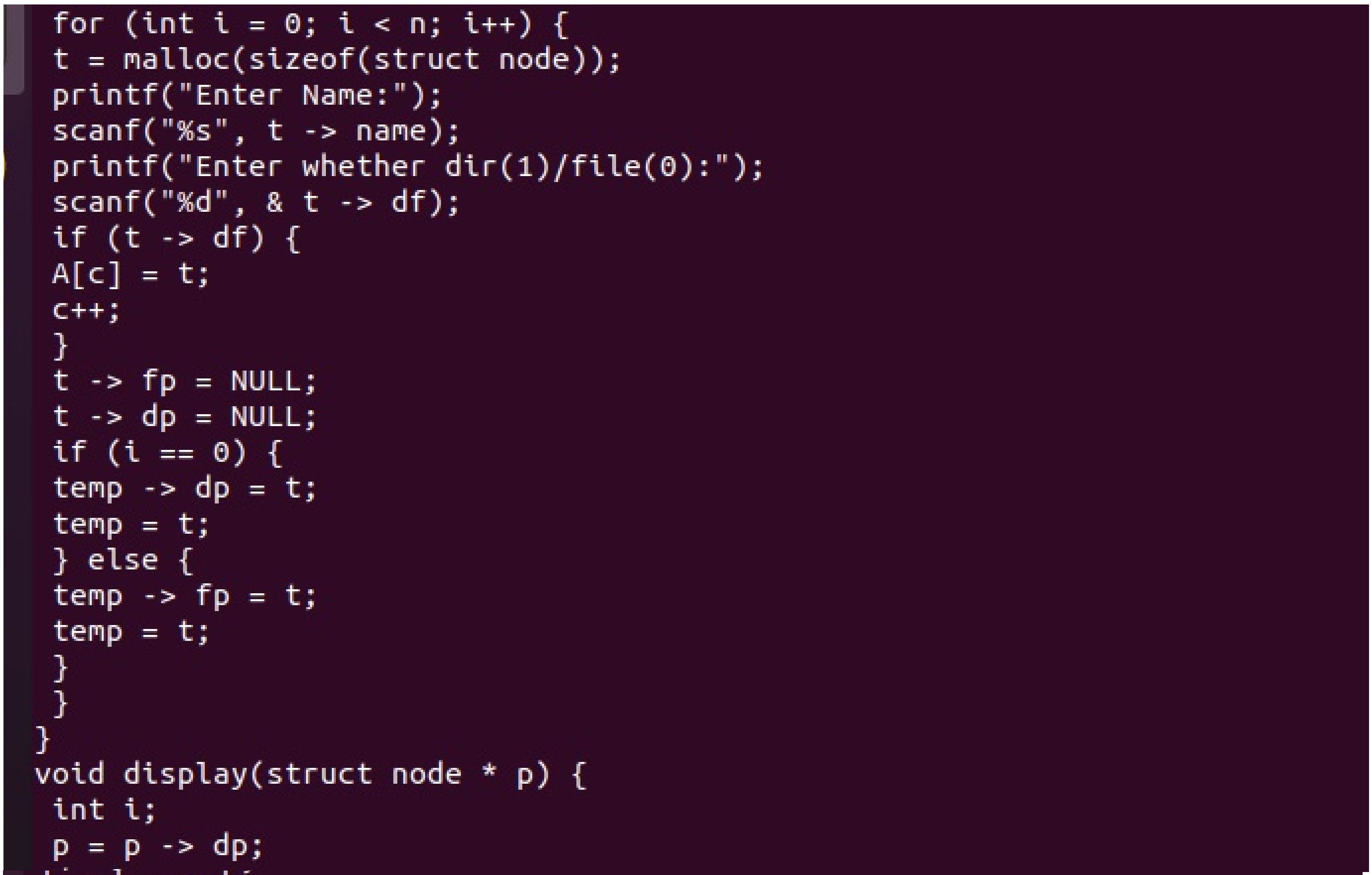
```
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 1

Enter name of directory -- dir1
Directory created
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 2

Enter name of the directory -- dir1
Enter name of the file -- file1
File created
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 2

Enter name of the directory -- dir1
Enter name of the file -- file
File created
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 3

Enter name of the directory -- dir1
Enter name of the file -- file1
File file1 not found
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 4

Enter name of the directory -- dir1
Enter the name of the file -- file
File file not found
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 5

Directory       Files
dir1
1. Create Directory     2. Create File  3. Delete File  4. Search File
Enter your choice -- 6
```

PROGRAM

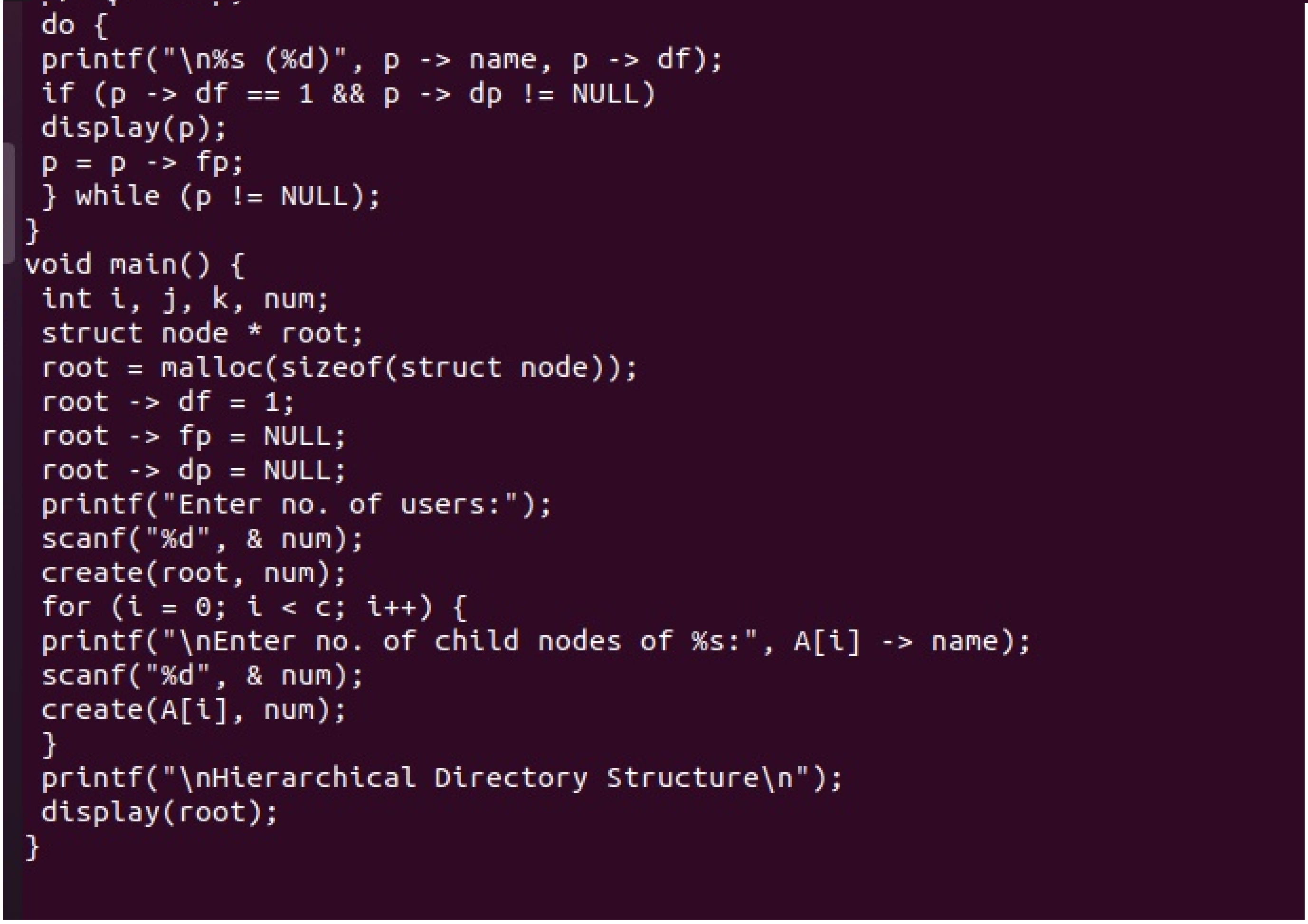
c) Hierarchical



```
#include<stdio.h>
#include<stdlib.h>
struct node {
 char name[25];
 int df;
 struct node * fp;
 struct node * dp;
};
struct node * A[20];
int c = 0;
void create(struct node * p, int n) {
 struct node * temp, * t;
 temp = p;
```

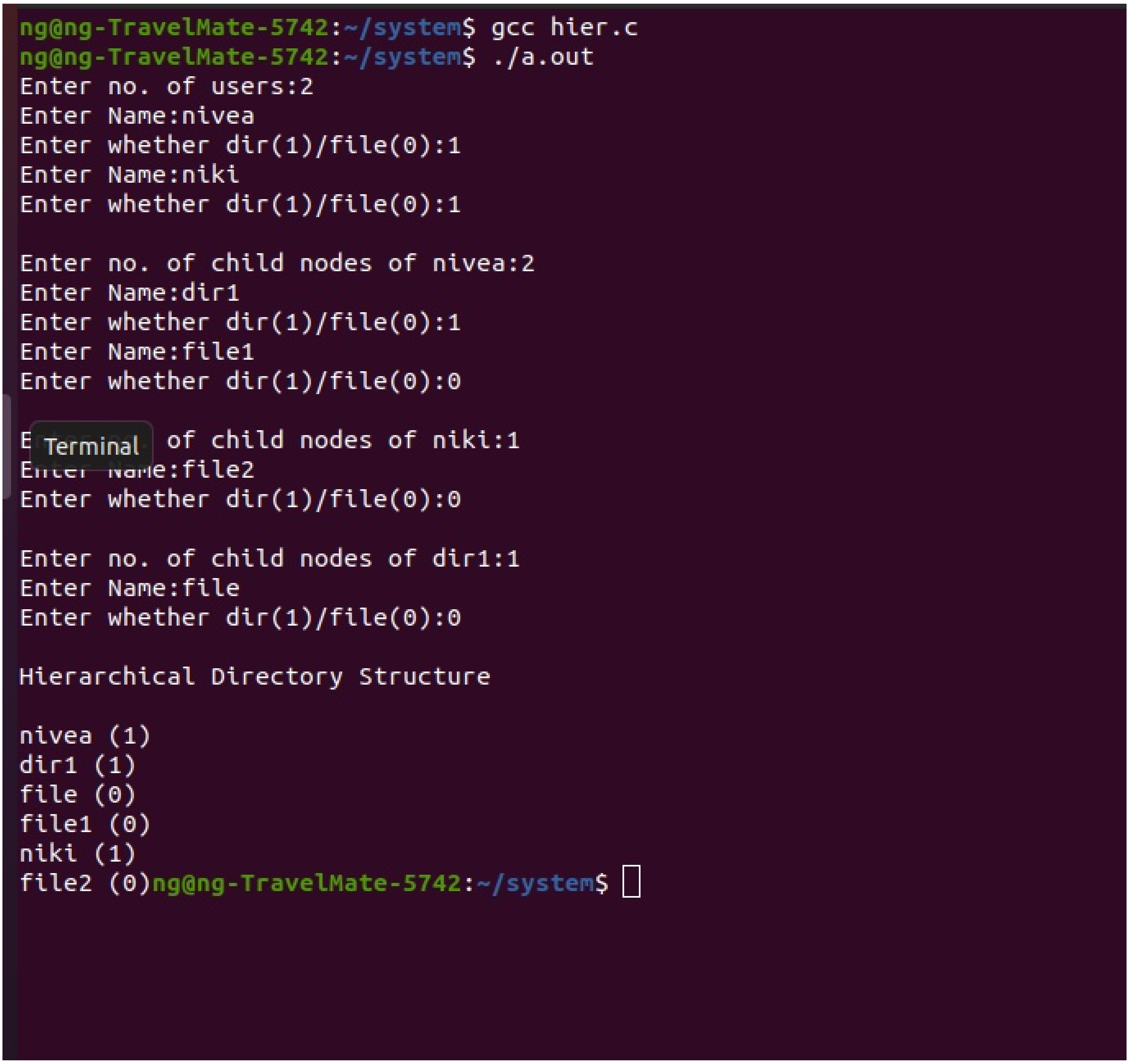


```
 for (int i = 0; i < n; i++) {
 t = malloc(sizeof(struct node));
 printf("Enter Name:");
 scanf("%s", t -> name);
 printf("Enter whether dir(1)/file(0):");
 scanf("%d", & t -> df);
 if (t -> df) {
 A[c] = t;
 c++;
 }
 t -> fp = NULL;
 t -> dp = NULL;
 if (i == 0) {
 temp -> dp = t;
 temp = t;
 } else {
 temp -> fp = t;
 temp = t;
 }
 }
}
void display(struct node * p) {
 int i;
 p = p -> dp;
```



```
 do {
 printf("\n%s (%d)", p -> name, p -> df);
 if (p -> df == 1 && p -> dp != NULL)
 display(p);
 p = p -> fp;
 } while (p != NULL);
}
void main() {
 int i, j, k, num;
 struct node * root;
 root = malloc(sizeof(struct node));
 root -> df = 1;
 root -> fp = NULL;
 root -> dp = NULL;
 printf("Enter no. of users:");
 scanf("%d", & num);
 create(root, num);
 for (i = 0; i < c; i++) {
 printf("\nEnter no. of child nodes of %s:", A[i] -> name);
 scanf("%d", & num);
 create(A[i], num);
 }
 printf("\nHierarchical Directory Structure\n");
 display(root);
}
```

OUTPUT



```
ng@ng-TravelMate-5742:~/system$ gcc hier.c
ng@ng-TravelMate-5742:~/system$ ./a.out
Enter no. of users:2
Enter Name:nivea
Enter whether dir(1)/file(0):1
Enter Name:niki
Enter whether dir(1)/file(0):1

Enter no. of child nodes of nivea:2
Enter Name:dir1
Enter whether dir(1)/file(0):1
Enter Name:file1
Enter whether dir(1)/file(0):0

Enter no. of child nodes of niki:1
Enter Name:file2
Enter whether dir(1)/file(0):0

Enter no. of child nodes of dir1:1
Enter Name:file
Enter whether dir(1)/file(0):0

Hierarchical Directory Structure

nivea (1)
dir1 (1)
file (0)
file1 (0)
niki (1)
file2 (0)ng@ng-TravelMate-5742:~/system$ 
```

RESULT

Program is executed successfully and output is obtained.

BY NIVEA GIGEN
S5 C
CHN18CS092