तमसो मा ज्योतिर्गमय

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

Innovative Assignment:

File Management System

Course Code and Name: 2CS301- Data Structures and Algorithms

Group Members:

Urvish Patel (21BCE221)

Utsav Patel (21BCE222)

Ved Patel (21BCE223)

Title:

File Management System

- OBJECTIVE:

The aim of the assignment to introduce students to the field of programming and uses of different types of data structures in real life.

In this projects we have implemented File Management System using different data structures like Tree, Circular doubly Linked List, Recursion , DFS (traversal) , etc. We have used C language to make File Management System.

- INTRODUCTION:

Everyone know how to use different types of gadgets in todays world but many of them don’t know how they works ,even many engineering student also don’t know . To design any software tool the main thing which is used is data structures. Data structures is very important to learn to design such a software tool also we have to think logically and have knowledge which data structure should be applied in particular tool and to access this data we need algorithms which must efficient in terms of time, space and power.

- Details of Project:

Our project title is File Management System. We all use it in our mobile phones, laptops. Here in this project we tried to implement it.

We have given file manager's functionality in our Project , Like (1)Add Folder and File, (2) Delete Folder and File (3) Copy and Paste Folder and File (4) Cut and Paste Folder and File (5) Search Folder and File

- Data Structure:

1) File: To store many files in the any the any given folder we have created a structure which contains file's information.
2) Folder: To store folders which store folder or file inside it we have created structure for the folder which itself contains two separate structure which is used to store files and folders inside a particular folder using circular doubly linked list.

- Functions:

(1) Add Folder and File :

We are adding folders and files by two methods.

(i) Adding folder and files in any folder.

With use of structure we have defined, we created a linked list in which all folders and files are stored for a particular folder and likewise for all other folder.

(ii) (ii)Adding folder and files to the Main drive (like we have C or D drive).
This function directly adds the folder or file to main drive.

(2) Delete Folder and File :

Delete folder:
This function is used to delete particular folder which is present in main memory or in any other folder.

Delete file:
This function is used to delete particular file which is present in main memory or in any other folder.

(3) Search Folder and Files :

This functionality is used which reduce the time of user to find any folder or file in the memory. If the folder of file is present then it will be shown to user otherwise it will give appropriate reply to user whether folder or file isn't present or deleted.

If the folder is present and found then it will show the all the folders and files contained by it.

If the file is present and found then it will the folder's information which contains this searched file (basically location of the file).

(4) Copy and Paste :

This functionality is used to store your data to any other folder. Here your data isn't delete from where you have copied instead it will assign to new memory block. It will use more memory in user device for any single data. It will copy any folder and paste it anywhere user wants it.

(5) Cut and Paste :

This functionality is similar to Copy and Paste but one difference is that in this feature we are assigning our data to some other place and delete the data from its old place.

➢ Details of different Data Structures & Algorithms used in our Assignment:

- **Tree:**

  We have used Tree data structure in our project. When we use file manager it creates a tree like structure to manage the files and folders information inside any folder. In this project files and folders act as a node and files are basically leaf nodes. And folders are internal nodes which are connected to other folders and files having more folders and files inside it like this it creates tree like structure.

- **Linked List:**

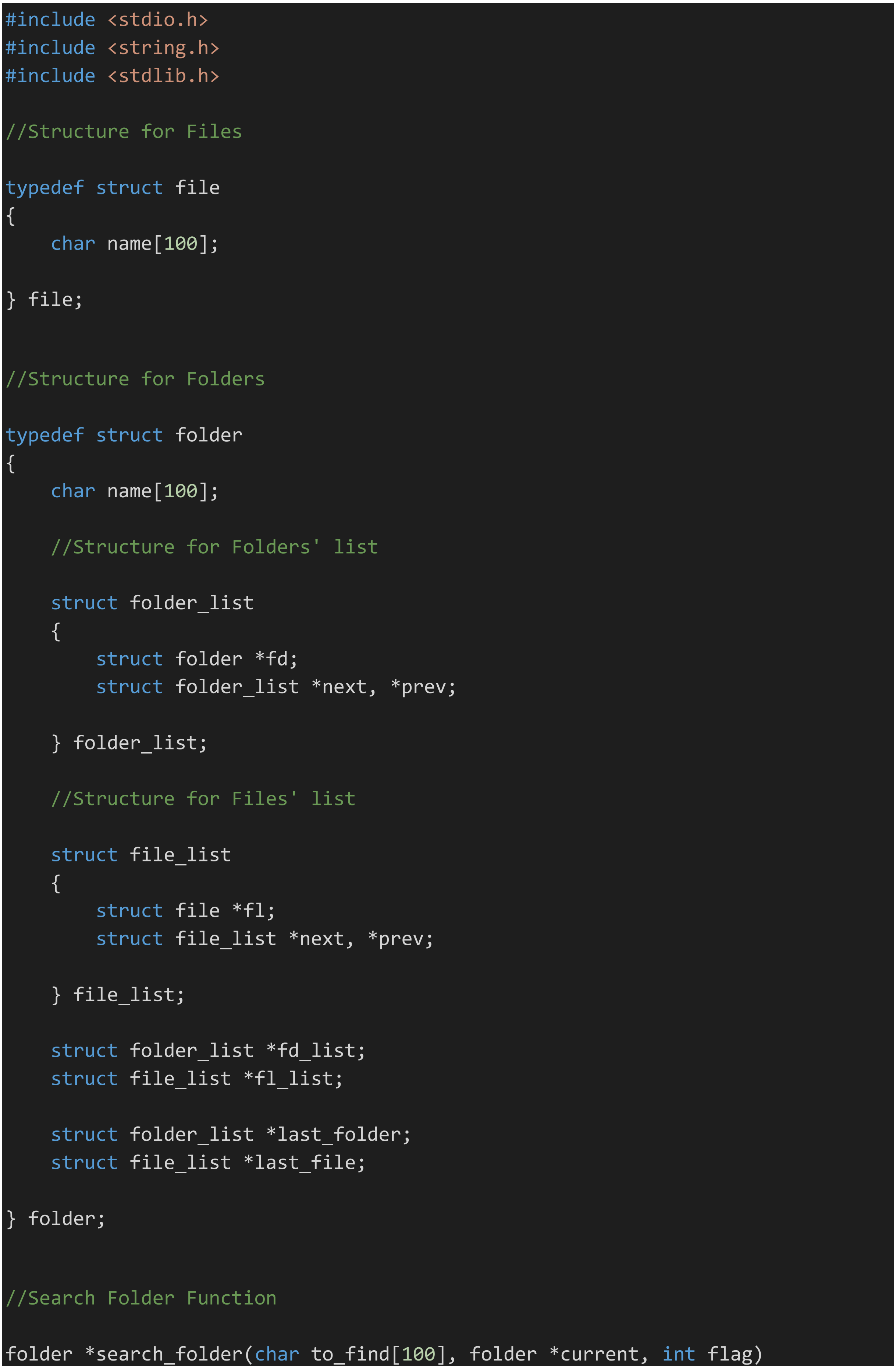
  We have used Circular Linked List to store data which is used for dynamic memory allocation. We dynamically created this to support variable number of files and folders.

- DFS:

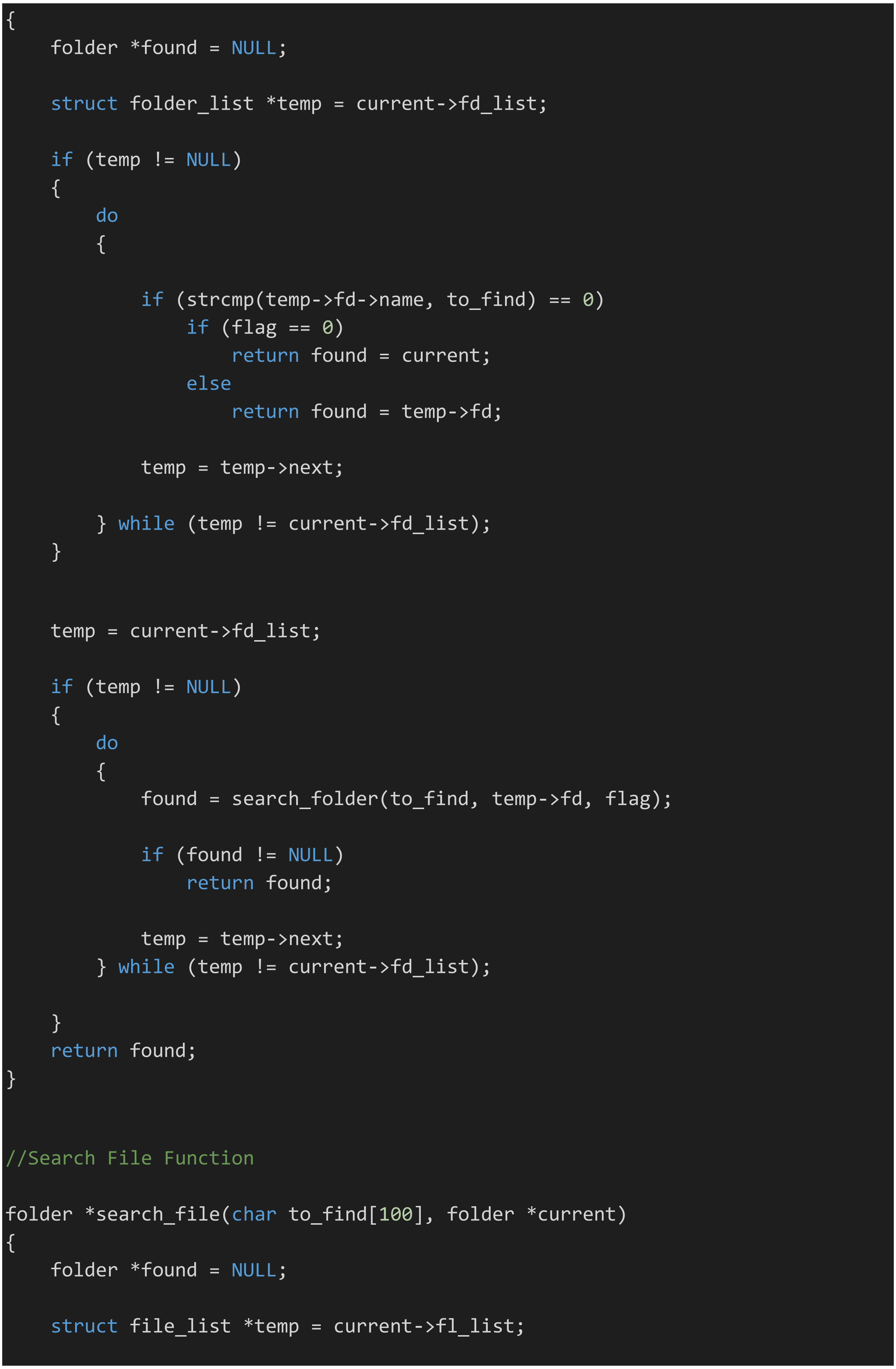
  When we want to delete folder or search any folder or file is in our list or not we search it using DFS (Depth First Search). This algorithm is used for the traversal of the tree like structure.

- Recursion:

  Recursion is used in delete and search function in DFS algorithm which will transfers all the node (files and folder) in the tree.

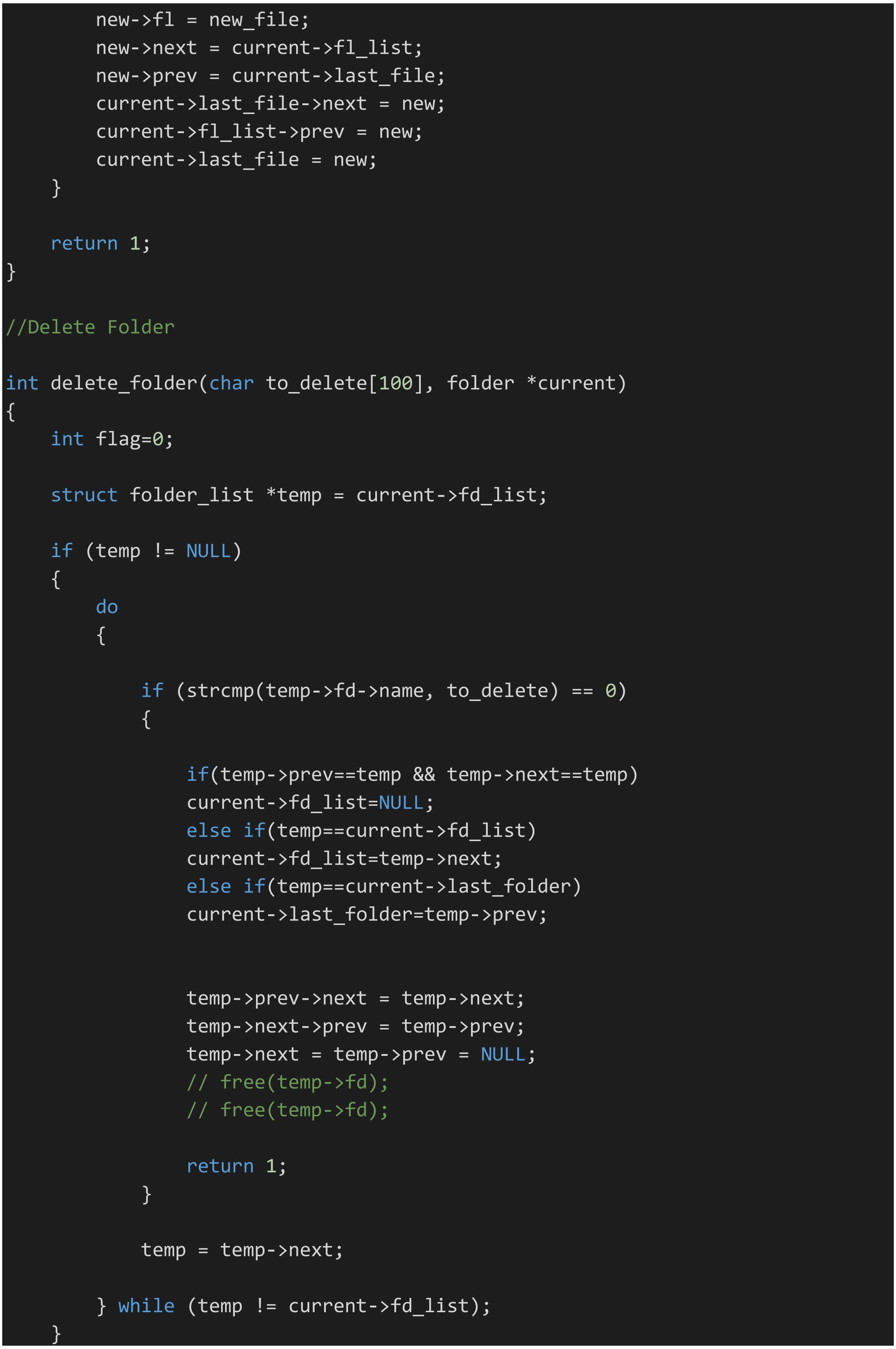


```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

//Structure for Files

typedef struct file
{
    char name[100];

} file;


//Structure for Folders

typedef struct folder
{
    char name[100];

    //Structure for Folders' list

    struct folder_list
    {
        struct folder *fd;
        struct folder_list *next, *prev;

    } folder_list;

    //Structure for Files' list

    struct file_list
    {
        struct file *fl;
        struct file_list *next, *prev;

    } file_list;

    struct folder_list *fd_list;
    struct file_list *fl_list;

    struct folder_list *last_folder;
    struct file_list *last_file;

} folder;


//Search Folder Function

folder *search_folder(char to_find[100], folder *current, int flag)
```

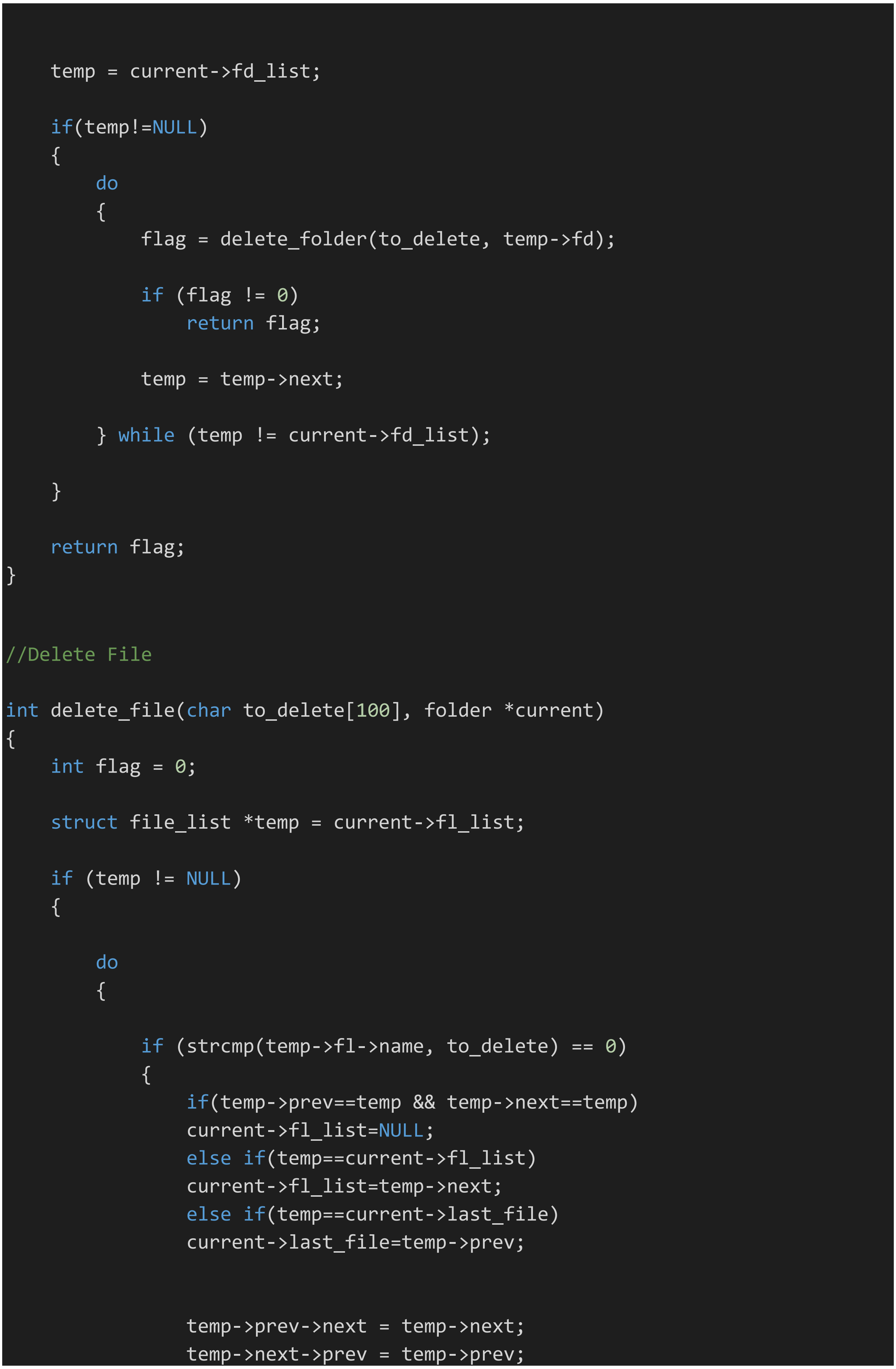


```
{
    folder *found = NULL;

    struct folder_list *temp = current->fd_list;

    if (temp != NULL)
    {
        do
        {

            if (strcmp(temp->fd->name, to_find) == 0)
                if (flag == 0)
                    return found = current;
                else
                    return found = temp->fd;

            temp = temp->next;

        } while (temp != current->fd_list);
    }


    temp = current->fd_list;

    if (temp != NULL)
    {
        do
        {
            found = search_folder(to_find, temp->fd, flag);

            if (found != NULL)
                return found;

            temp = temp->next;
        } while (temp != current->fd_list);

    }
    return found;
}


//Search File Function

folder *search_file(char to_find[100], folder *current)
{
    folder *found = NULL;

    struct file_list *temp = current->fl_list;
    if (temp != NULL)
    {
        do
        {

            if (strcmp(temp->fl->name, to_find) == 0)
                return found = current;

            temp = temp->next;
        } while (temp != current->fl_list);
    }


    struct folder_list *list = current->fd_list;


    if (list != NULL)
    {
        do
        {
            found = search_file(to_find, list->fd);

            if (found != NULL)
                return found;

            list = list->next;

        } while (list != current->fd_list);



    }

    return found;
}


//Add a folder

int add_folder(char name[100], folder *current)
{
    folder *new_folder = (folder *)calloc(1, sizeof(folder));

    strcpy(new_folder->name, name);

    struct folder_list *new = (struct folder_list *)calloc(1, sizeof(struct
folder_list));

    if (new_folder == NULL || new == NULL)
```

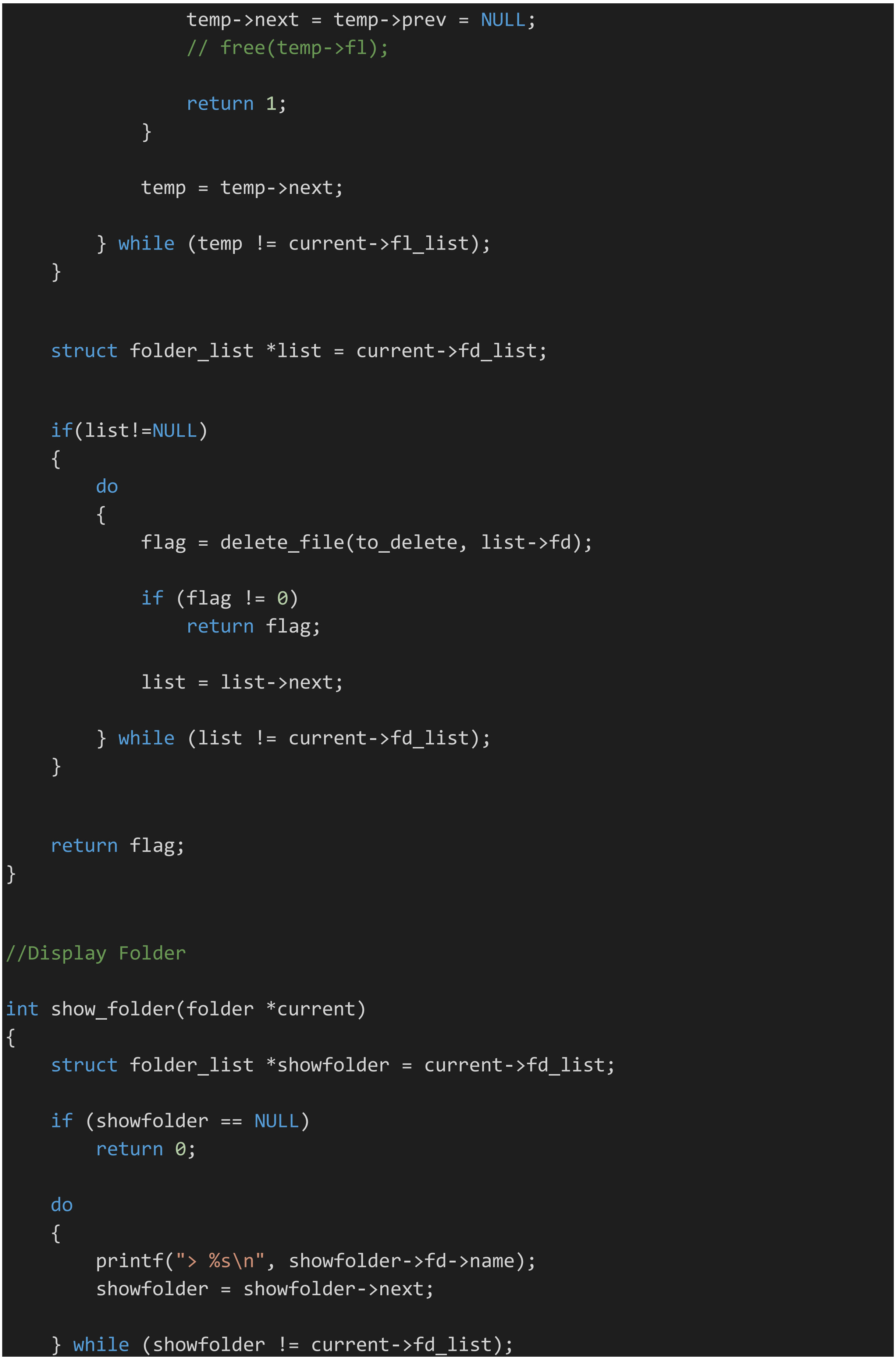
```
        return 0;

    if (current->fd_list == NULL)
    {
        new->fd = new_folder;
        new->next = new;
        new->prev = new;

        current->fd_list = new;
        current->last_folder = new;
    }
    else
    {
        new->fd = new_folder;
        new->next = current->fd_list;
        new->prev = current->last_folder;
        current->last_folder->next = new;
        current->fd_list->prev = new;
        current->last_folder = new;
    }

    return 1;
}


//Add a file

int add_file(char name[100], folder *current)
{
    file *new_file = (file *)calloc(1, sizeof(file));

    strcpy(new_file->name, name);

    struct file_list *new = (struct file_list *)calloc(1, sizeof(struct
file_list));

    if (new_file == NULL || new == NULL)
        return 0;

    if (current->fl_list == NULL)
    {
        new->fl = new_file;
        new->next = new;
        new->prev = new;
        current->fl_list = new;
        current->last_file = new;
    }
    else
    {
```



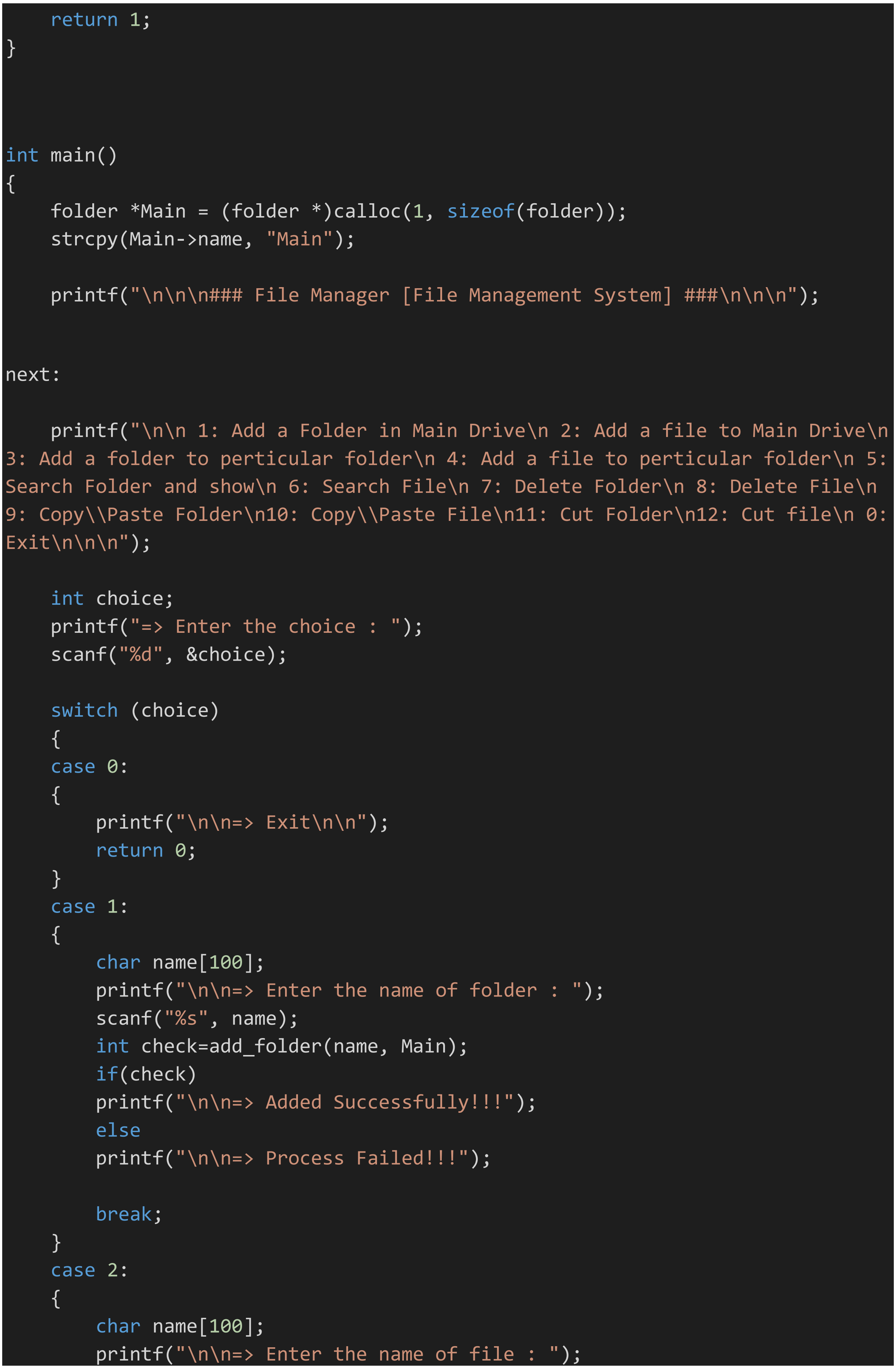
```
        new->fl = new_file;
        new->next = current->fl_list;
        new->prev = current->last_file;
        current->last_file->next = new;
        current->fl_list->prev = new;
        current->last_file = new;
    }

    return 1;
}

//Delete Folder

int delete_folder(char to_delete[100], folder *current)
{
    int flag=0;

    struct folder_list *temp = current->fd_list;

    if (temp != NULL)
    {
        do
        {

            if (strcmp(temp->fd->name, to_delete) == 0)
            {

                if(temp->prev==temp && temp->next==temp)
                current->fd_list=NULL;
                else if(temp==current->fd_list)
                current->fd_list=temp->next;
                else if(temp==current->last_folder)
                current->last_folder=temp->prev;


                temp->prev->next = temp->next;
                temp->next->prev = temp->prev;
                temp->next = temp->prev = NULL;
                // free(temp->fd);
                // free(temp->fd);

                return 1;
            }

            temp = temp->next;

        } while (temp != current->fd_list);
    }
```



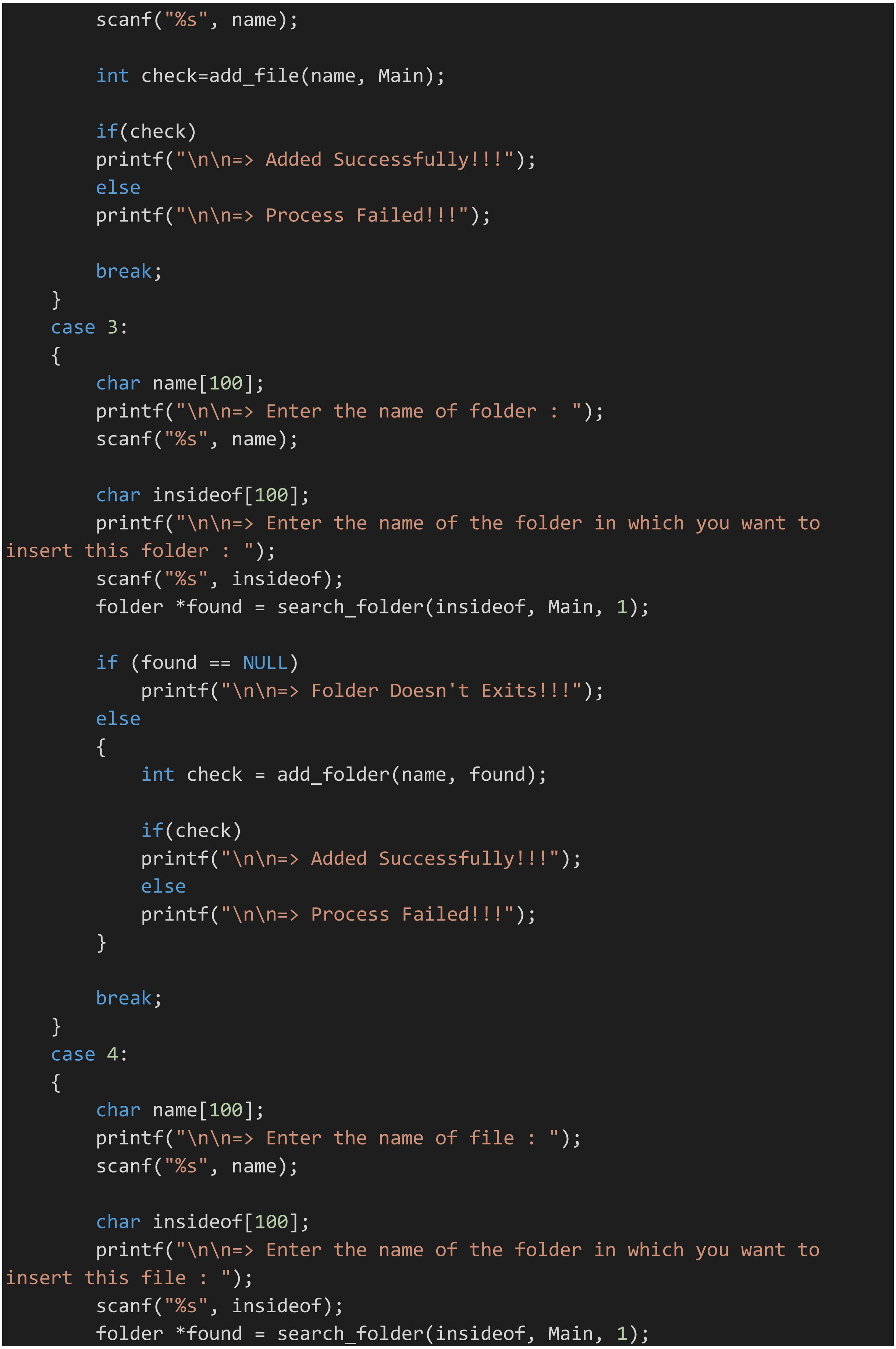
```
    temp = current->fd_list;

    if(temp!=NULL)
    {
        do
        {
            flag = delete_folder(to_delete, temp->fd);

            if (flag != 0)
                return flag;

            temp = temp->next;

        } while (temp != current->fd_list);

    }

    return flag;
}


//Delete File

int delete_file(char to_delete[100], folder *current)
{
    int flag = 0;

    struct file_list *temp = current->fl_list;

    if (temp != NULL)
    {

        do
        {

            if (strcmp(temp->fl->name, to_delete) == 0)
            {
                if(temp->prev==temp && temp->next==temp)
                current->fl_list=NULL;
                else if(temp==current->fl_list)
                current->fl_list=temp->next;
                else if(temp==current->last_file)
                current->last_file=temp->prev;


                temp->prev->next = temp->next;
                temp->next->prev = temp->prev;
```



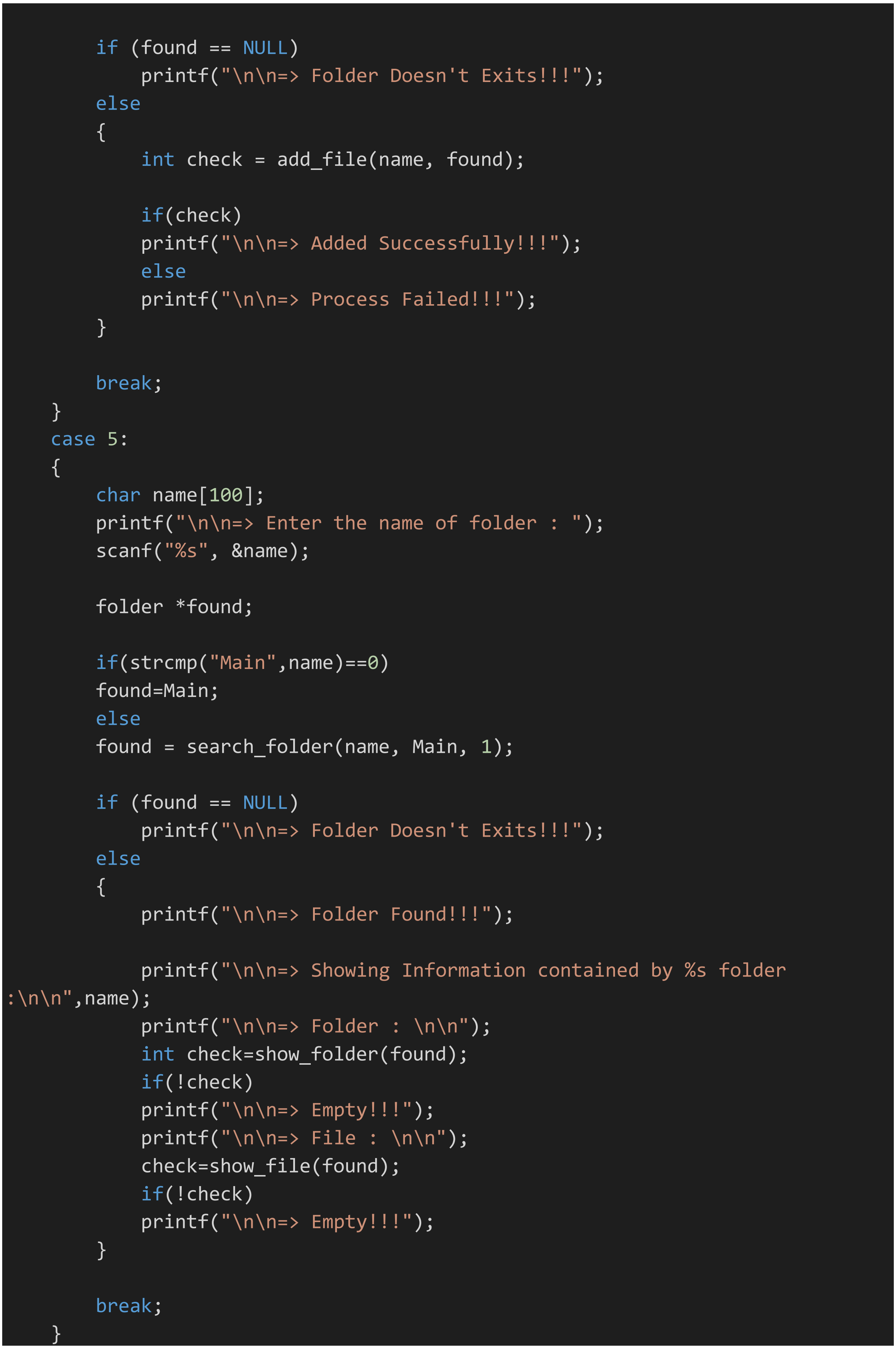
```
                temp->next = temp->prev = NULL;
                // free(temp->fl);

                return 1;
            }

            temp = temp->next;

        } while (temp != current->fl_list);
    }


    struct folder_list *list = current->fd_list;


    if(list!=NULL)
    {
        do
        {
            flag = delete_file(to_delete, list->fd);

            if (flag != 0)
                return flag;

            list = list->next;

        } while (list != current->fd_list);
    }


    return flag;
}


//Display Folder

int show_folder(folder *current)
{
    struct folder_list *showfolder = current->fd_list;

    if (showfolder == NULL)
        return 0;

    do
    {
        printf("> %s\n", showfolder->fd->name);
        showfolder = showfolder->next;

    } while (showfolder != current->fd_list);
    return 1;
}


//Display File

int show_file(folder *current)
{
    struct file_list *showfile = current->fl_list;

    if (showfile == NULL)
        return 0;

    do
    {
        printf("> %s\n", showfile->fl->name);
        showfile = showfile->next;
    } while (showfile != current->fl_list);

    return 1;
}

int copy_folder(folder *copy, folder *current)
{
    struct folder_list *new = (struct folder_list *)calloc(1, sizeof(struct folder_list));

    if (new == NULL)
        return 0;

    if (current->fd_list == NULL)
    {
        new->fd = copy;
        new->next = new;
        new->prev = new;
        current->fd_list = new;
        current->last_folder = new;
    }
    else
    {
        new->fd = copy;
        new->next = current->fd_list;
        new->prev = current->last_folder;
        current->last_folder->next = new;
        current->fd_list->prev = new;
        current->last_folder = new;
    }
```



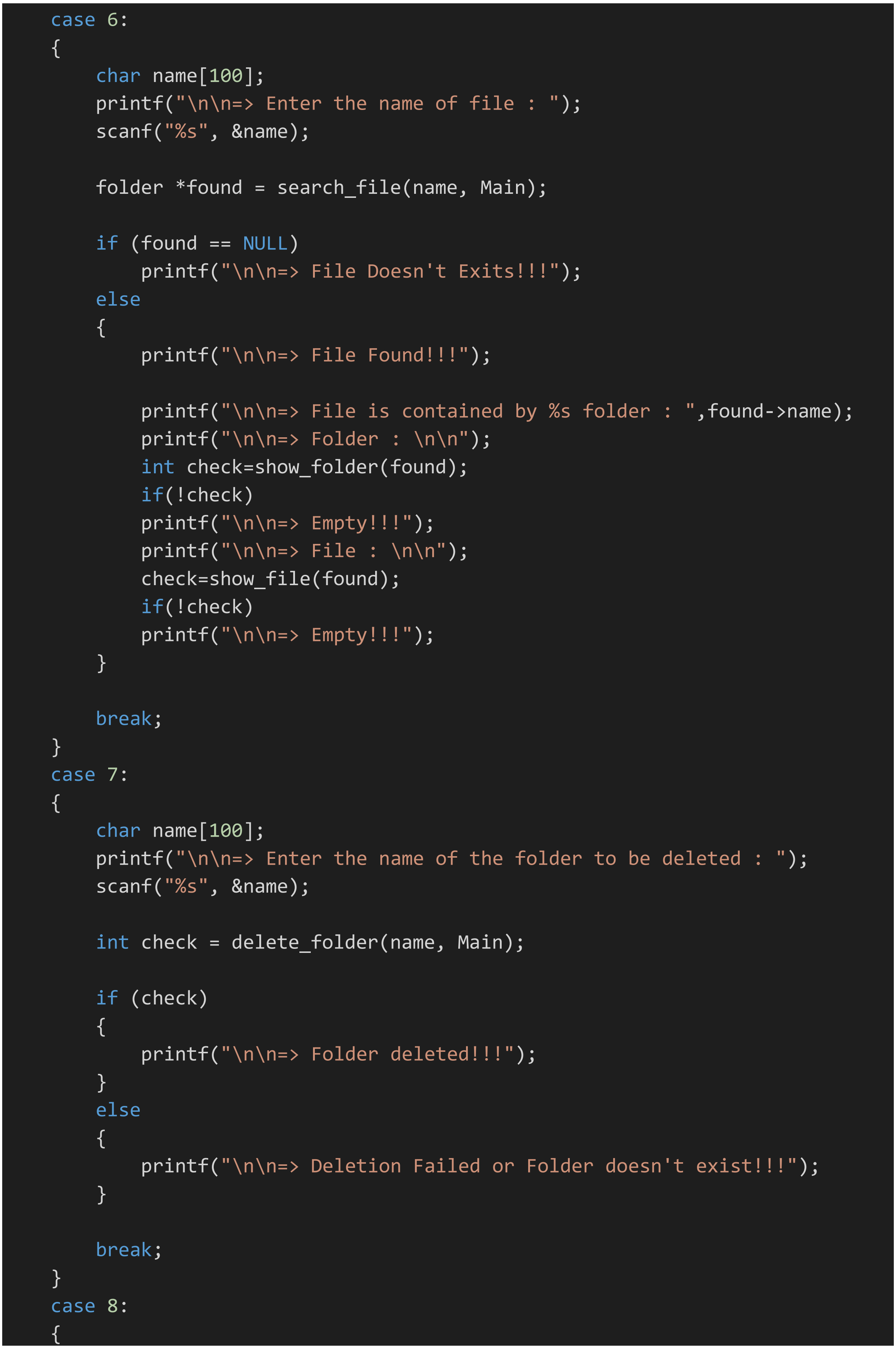
```
    return 1;
}



int main()
{
    folder *Main = (folder *)calloc(1, sizeof(folder));
    strcpy(Main->name, "Main");

    printf("\n\n\n### File Manager [File Management System] ###\n\n\n");


next:

    printf("\n\n 1: Add a Folder in Main Drive\n 2: Add a file to Main Drive\n
3: Add a folder to perticular folder\n 4: Add a file to perticular folder\n 5:
Search Folder and show\n 6: Search File\n 7: Delete Folder\n 8: Delete File\n
9: Copy\\Paste Folder\n10: Copy\\Paste File\n11: Cut Folder\n12: Cut file\n 0:
Exit\n\n\n");

    int choice;
    printf("=> Enter the choice : ");
    scanf("%d", &choice);

    switch (choice)
    {
    case 0:
    {
        printf("\n\n=> Exit\n\n");
        return 0;
    }
    case 1:
    {
        char name[100];
        printf("\n\n=> Enter the name of folder : ");
        scanf("%s", name);
        int check=add_folder(name, Main);
        if(check)
        printf("\n\n=> Added Successfully!!!");
        else
        printf("\n\n=> Process Failed!!!");

        break;
    }
    case 2:
    {
        char name[100];
        printf("\n\n=> Enter the name of file : ");
```



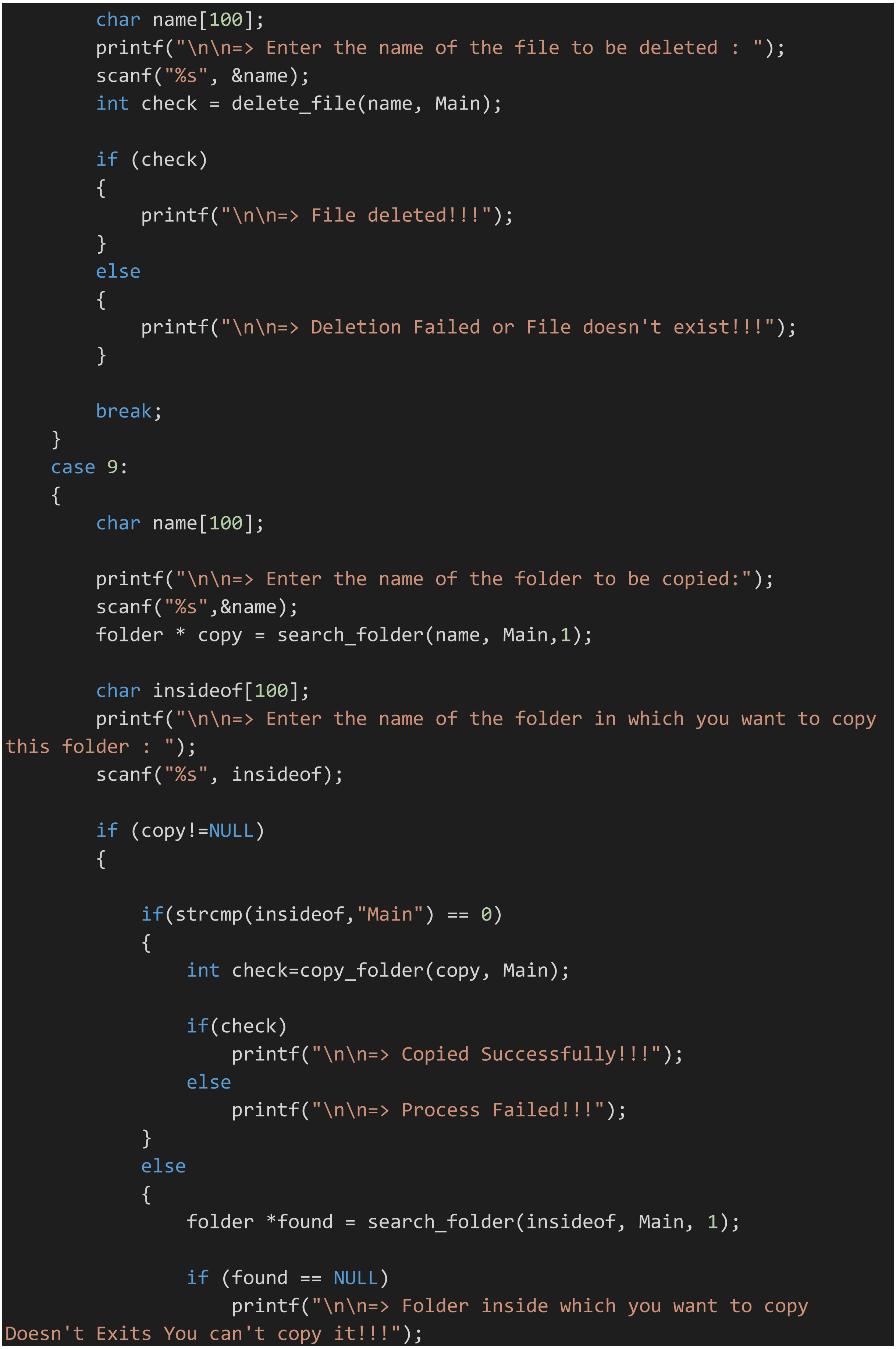
```
        scanf("%s", name);

        int check=add_file(name, Main);

        if(check)
        printf("\n\n=> Added Successfully!!!");
        else
        printf("\n\n=> Process Failed!!!");

        break;
    }
    case 3:
    {
        char name[100];
        printf("\n\n=> Enter the name of folder : ");
        scanf("%s", name);

        char insideof[100];
        printf("\n\n=> Enter the name of the folder in which you want to
insert this folder : ");
        scanf("%s", insideof);
        folder *found = search_folder(insideof, Main, 1);

        if (found == NULL)
            printf("\n\n=> Folder Doesn't Exits!!!");
        else
        {
            int check = add_folder(name, found);

            if(check)
            printf("\n\n=> Added Successfully!!!");
            else
            printf("\n\n=> Process Failed!!!");
        }

        break;
    }
    case 4:
    {
        char name[100];
        printf("\n\n=> Enter the name of file : ");
        scanf("%s", name);

        char insideof[100];
        printf("\n\n=> Enter the name of the folder in which you want to
insert this file : ");
        scanf("%s", insideof);
        folder *found = search_folder(insideof, Main, 1);
```



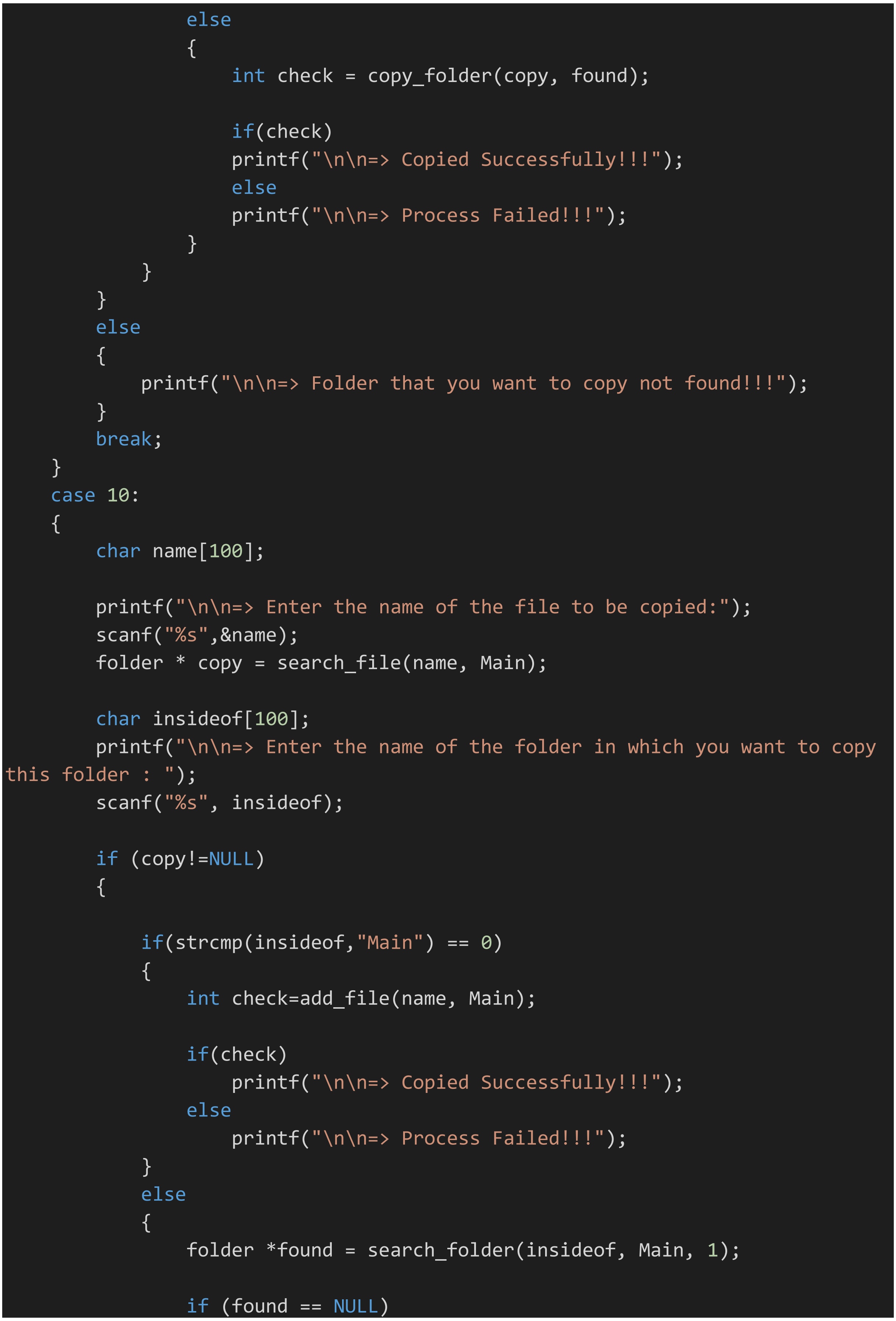
```
        if (found == NULL)
            printf("\n\n=> Folder Doesn't Exits!!!");
        else
        {
            int check = add_file(name, found);

            if(check)
            printf("\n\n=> Added Successfully!!!");
            else
            printf("\n\n=> Process Failed!!!");
        }

        break;
    }
    case 5:
    {
        char name[100];
        printf("\n\n=> Enter the name of folder : ");
        scanf("%s", &name);

        folder *found;

        if(strcmp("Main",name)==0)
        found=Main;
        else
        found = search_folder(name, Main, 1);

        if (found == NULL)
            printf("\n\n=> Folder Doesn't Exits!!!");
        else
        {
            printf("\n\n=> Folder Found!!!");

            printf("\n\n=> Showing Information contained by %s folder
:\n\n",name);
            printf("\n\n=> Folder : \n\n");
            int check=show_folder(found);
            if(!check)
            printf("\n\n=> Empty!!!");
            printf("\n\n=> File : \n\n");
            check=show_file(found);
            if(!check)
            printf("\n\n=> Empty!!!");
        }

        break;
    }
```



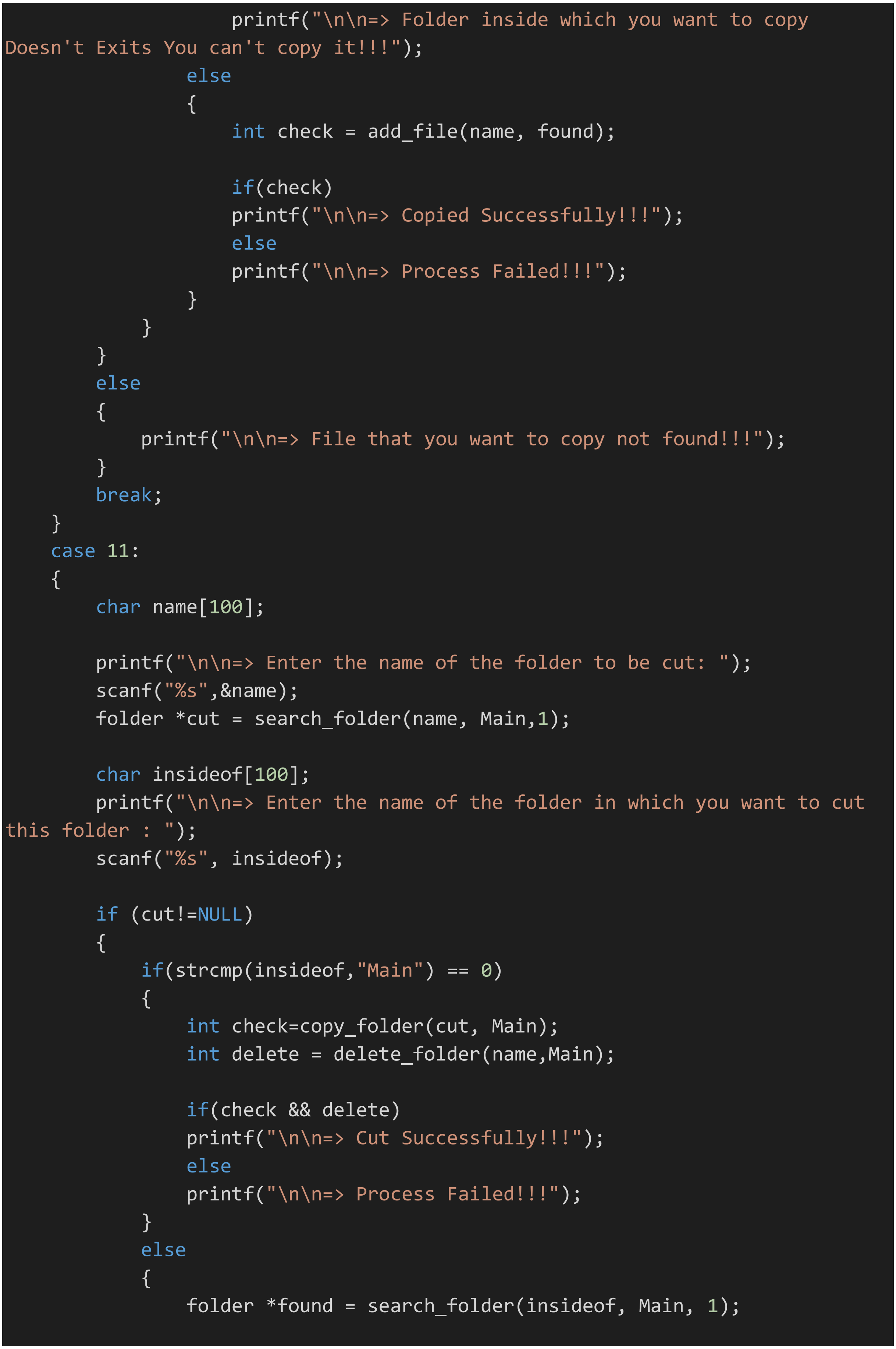
```
    case 6:
    {
        char name[100];
        printf("\n\n=> Enter the name of file : ");
        scanf("%s", &name);

        folder *found = search_file(name, Main);

        if (found == NULL)
            printf("\n\n=> File Doesn't Exits!!!");
        else
        {
            printf("\n\n=> File Found!!!");

            printf("\n\n=> File is contained by %s folder : ",found->name);
            printf("\n\n=> Folder : \n\n");
            int check=show_folder(found);
            if(!check)
            printf("\n\n=> Empty!!!");
            printf("\n\n=> File : \n\n");
            check=show_file(found);
            if(!check)
            printf("\n\n=> Empty!!!");
        }

        break;
    }
    case 7:
    {
        char name[100];
        printf("\n\n=> Enter the name of the folder to be deleted : ");
        scanf("%s", &name);

        int check = delete_folder(name, Main);

        if (check)
        {
            printf("\n\n=> Folder deleted!!!");
        }
        else
        {
            printf("\n\n=> Deletion Failed or Folder doesn't exist!!!");
        }

        break;
    }
    case 8:
    {
```



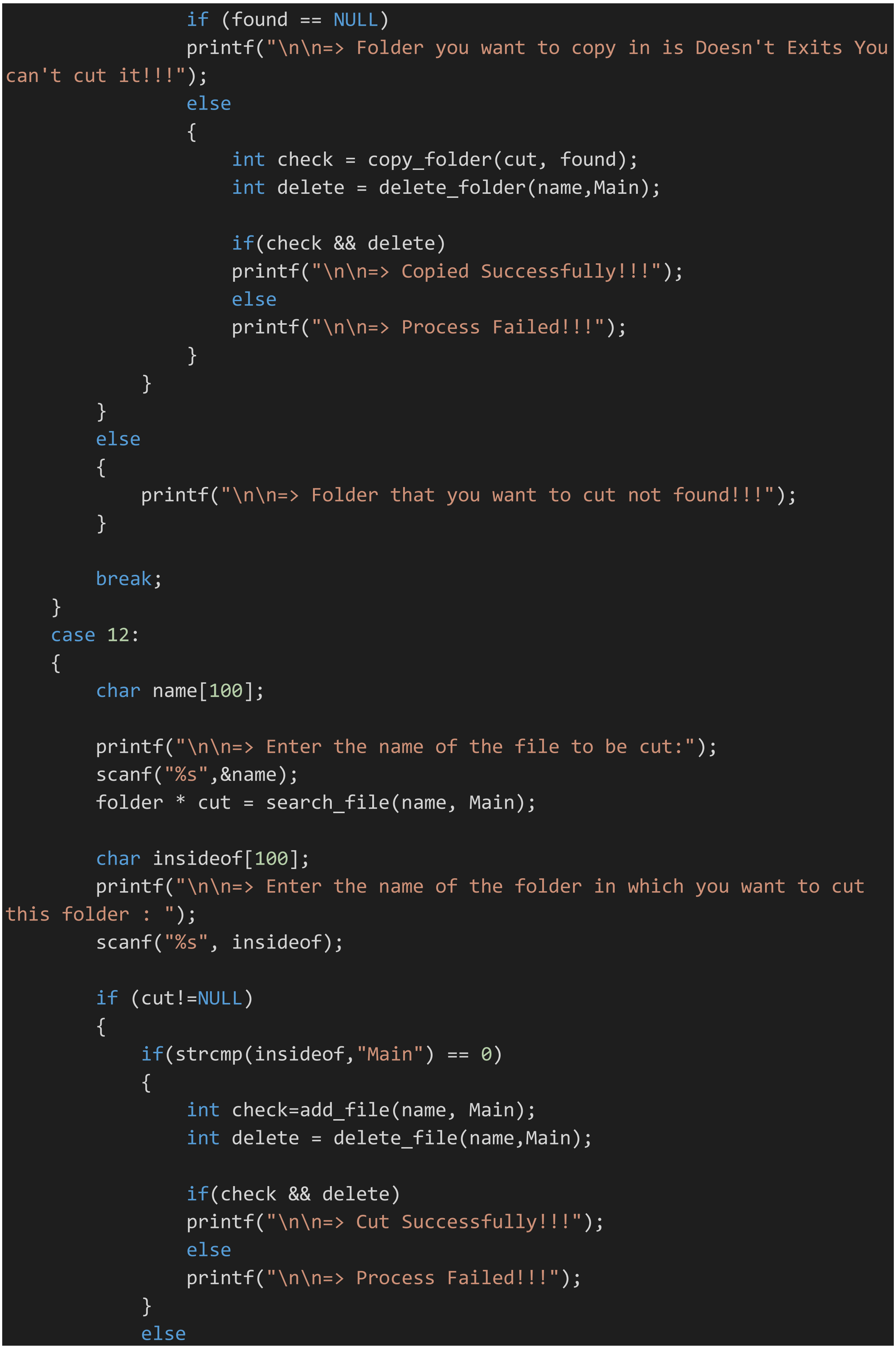
```
        char name[100];
        printf("\n\n=> Enter the name of the file to be deleted : ");
        scanf("%s", &name);
        int check = delete_file(name, Main);

        if (check)
        {
            printf("\n\n=> File deleted!!!");
        }
        else
        {
            printf("\n\n=> Deletion Failed or File doesn't exist!!!");
        }

        break;
    }
    case 9:
    {
        char name[100];

        printf("\n\n=> Enter the name of the folder to be copied:");
        scanf("%s",&name);
        folder * copy = search_folder(name, Main,1);

        char insideof[100];
        printf("\n\n=> Enter the name of the folder in which you want to copy
this folder : ");
        scanf("%s", insideof);

        if (copy!=NULL)
        {

            if(strcmp(insideof,"Main") == 0)
            {
                int check=copy_folder(copy, Main);

                if(check)
                    printf("\n\n=> Copied Successfully!!!");
                else
                    printf("\n\n=> Process Failed!!!");
            }
            else
            {
                folder *found = search_folder(insideof, Main, 1);

                if (found == NULL)
                    printf("\n\n=> Folder inside which you want to copy
Doesn't Exits You can't copy it!!!");
```



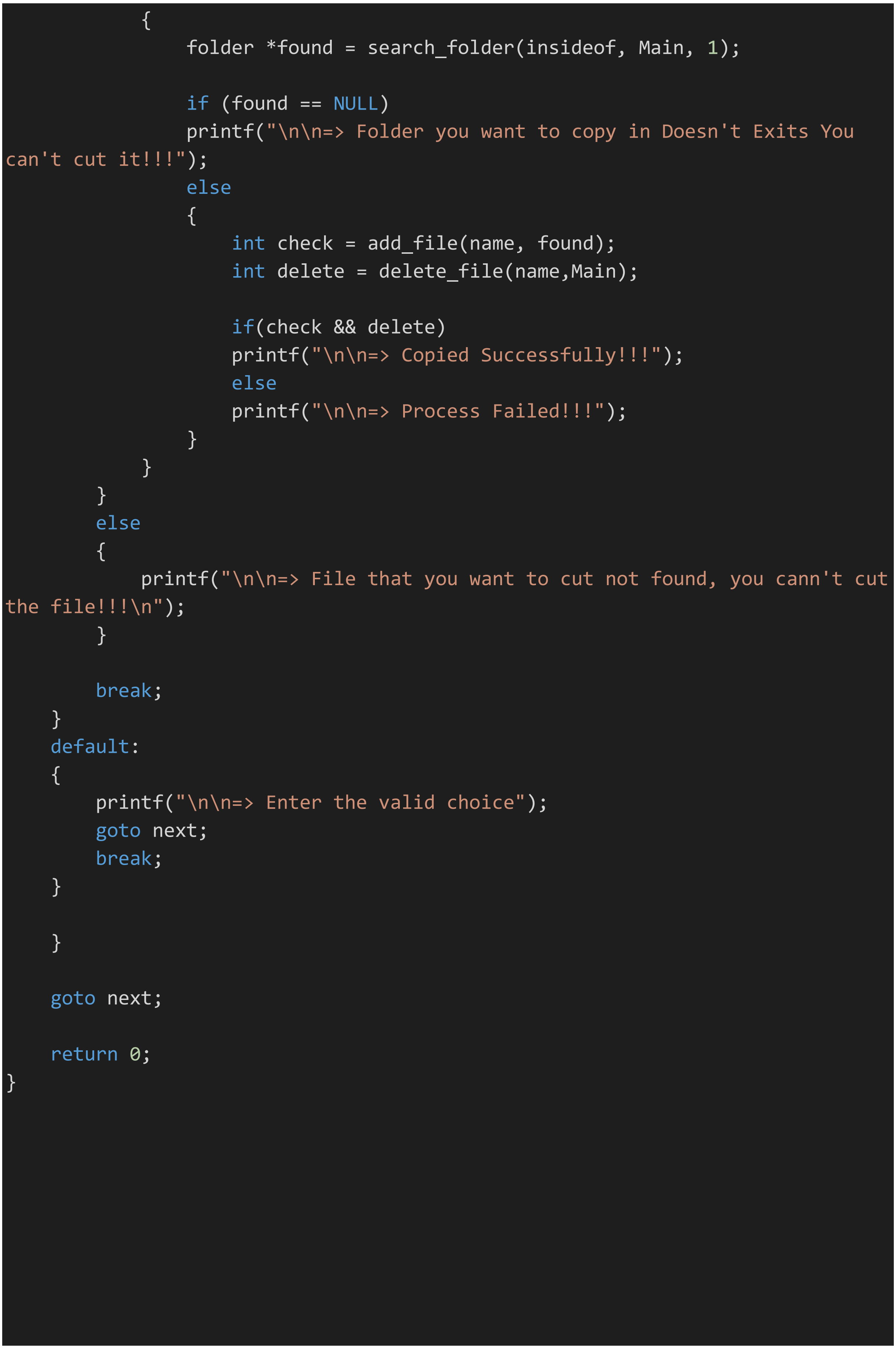
```
                else
                {
                    int check = copy_folder(copy, found);

                    if(check)
                    printf("\n\n=> Copied Successfully!!!");
                    else
                    printf("\n\n=> Process Failed!!!");
                }
            }
        }
        else
        {
            printf("\n\n=> Folder that you want to copy not found!!!");
        }
        break;
    }
    case 10:
    {
        char name[100];

        printf("\n\n=> Enter the name of the file to be copied:");
        scanf("%s",&name);
        folder * copy = search_file(name, Main);

        char insideof[100];
        printf("\n\n=> Enter the name of the folder in which you want to copy
this folder : ");
        scanf("%s", insideof);

        if (copy!=NULL)
        {

            if(strcmp(insideof,"Main") == 0)
            {
                int check=add_file(name, Main);

                if(check)
                    printf("\n\n=> Copied Successfully!!!");
                else
                    printf("\n\n=> Process Failed!!!");
            }
            else
            {
                folder *found = search_folder(insideof, Main, 1);

                if (found == NULL)
```



```
                    printf("\n\n=> Folder inside which you want to copy
Doesn't Exits You can't copy it!!!");
                else
                {
                    int check = add_file(name, found);

                    if(check)
                    printf("\n\n=> Copied Successfully!!!");
                    else
                    printf("\n\n=> Process Failed!!!");
                }
            }
        }
        else
        {
            printf("\n\n=> File that you want to copy not found!!!");
        }
        break;
    }
    case 11:
    {
        char name[100];

        printf("\n\n=> Enter the name of the folder to be cut: ");
        scanf("%s",&name);
        folder *cut = search_folder(name, Main,1);

        char insideof[100];
        printf("\n\n=> Enter the name of the folder in which you want to cut
this folder : ");
        scanf("%s", insideof);

        if (cut!=NULL)
        {
            if(strcmp(insideof,"Main") == 0)
            {
                int check=copy_folder(cut, Main);
                int delete = delete_folder(name,Main);

                if(check && delete)
                printf("\n\n=> Cut Successfully!!!");
                else
                printf("\n\n=> Process Failed!!!");
            }
            else
            {
                folder *found = search_folder(insideof, Main, 1);
```

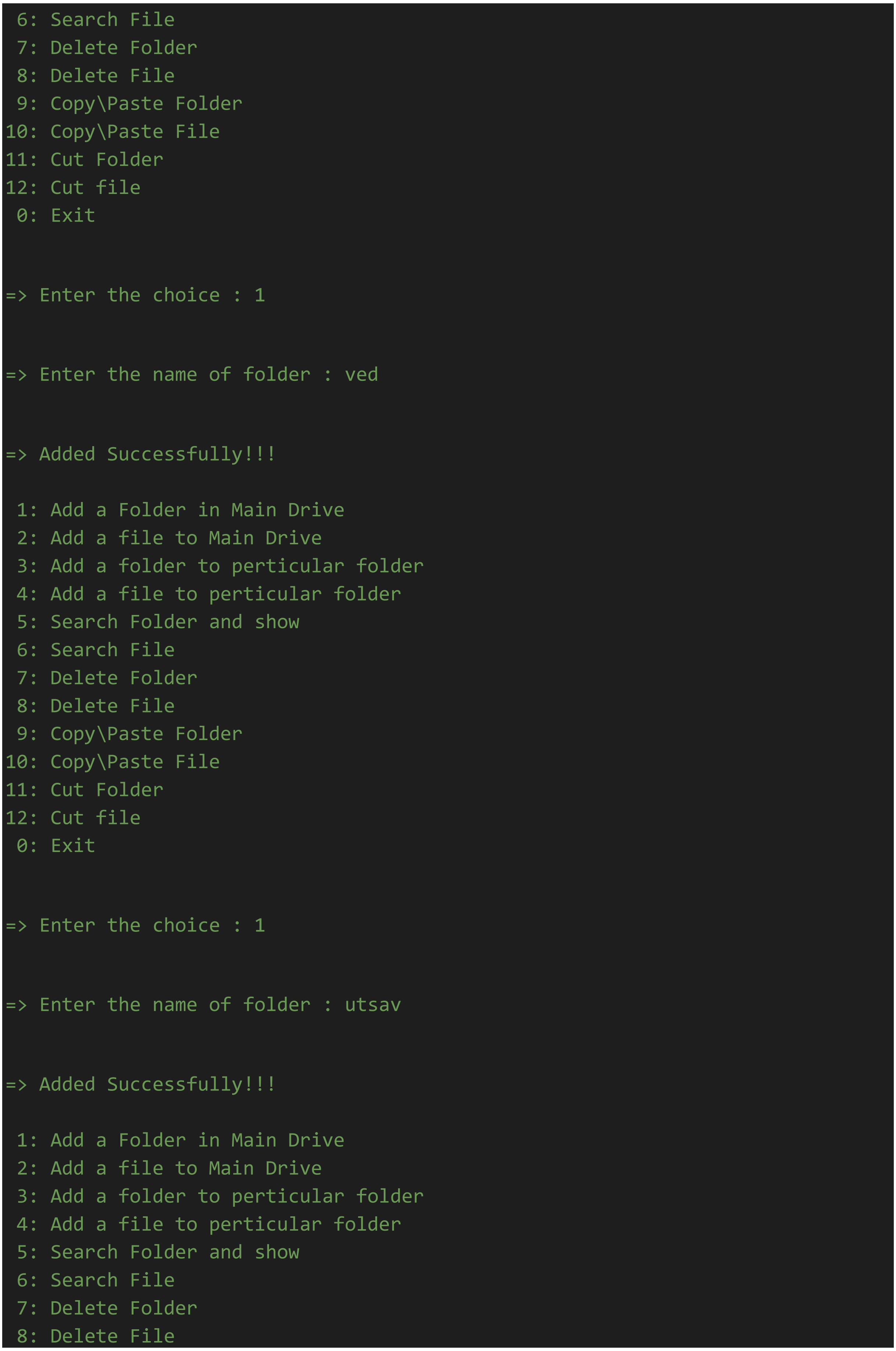


```
                if (found == NULL)
                printf("\n\n=> Folder you want to copy in is Doesn't Exits You
can't cut it!!!");
                else
                {
                    int check = copy_folder(cut, found);
                    int delete = delete_folder(name,Main);

                    if(check && delete)
                    printf("\n\n=> Copied Successfully!!!");
                    else
                    printf("\n\n=> Process Failed!!!");
                }
            }
        }
        else
        {
            printf("\n\n=> Folder that you want to cut not found!!!");
        }

        break;
    }
    case 12:
    {
        char name[100];

        printf("\n\n=> Enter the name of the file to be cut:");
        scanf("%s",&name);
        folder * cut = search_file(name, Main);

        char insideof[100];
        printf("\n\n=> Enter the name of the folder in which you want to cut
this folder : ");
        scanf("%s", insideof);

        if (cut!=NULL)
        {
            if(strcmp(insideof,"Main") == 0)
            {
                int check=add_file(name, Main);
                int delete = delete_file(name,Main);

                if(check && delete)
                printf("\n\n=> Cut Successfully!!!");
                else
                printf("\n\n=> Process Failed!!!");
            }
            else
```

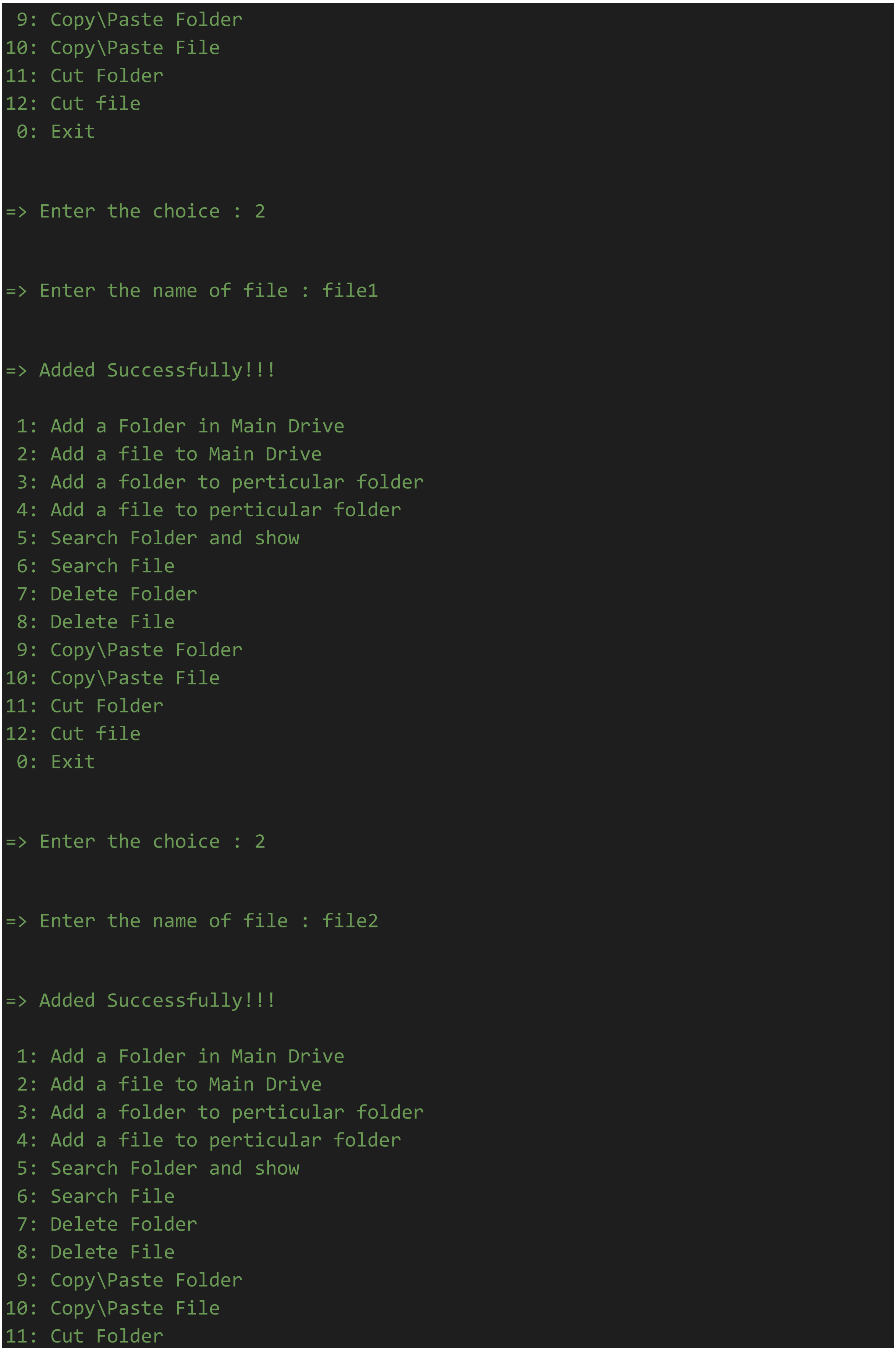


```
            {
                folder *found = search_folder(insideof, Main, 1);

                if (found == NULL)
                printf("\n\n=> Folder you want to copy in Doesn't Exits You
can't cut it!!!");
                else
                {
                    int check = add_file(name, found);
                    int delete = delete_file(name,Main);

                    if(check && delete)
                    printf("\n\n=> Copied Successfully!!!");
                    else
                    printf("\n\n=> Process Failed!!!");
                }
            }
        }
        else
        {
            printf("\n\n=> File that you want to cut not found, you cann't cut
the file!!!\n");
        }

        break;
    }
    default:
    {
        printf("\n\n=> Enter the valid choice");
        goto next;
        break;
    }

    }

    goto next;

    return 0;
}
```

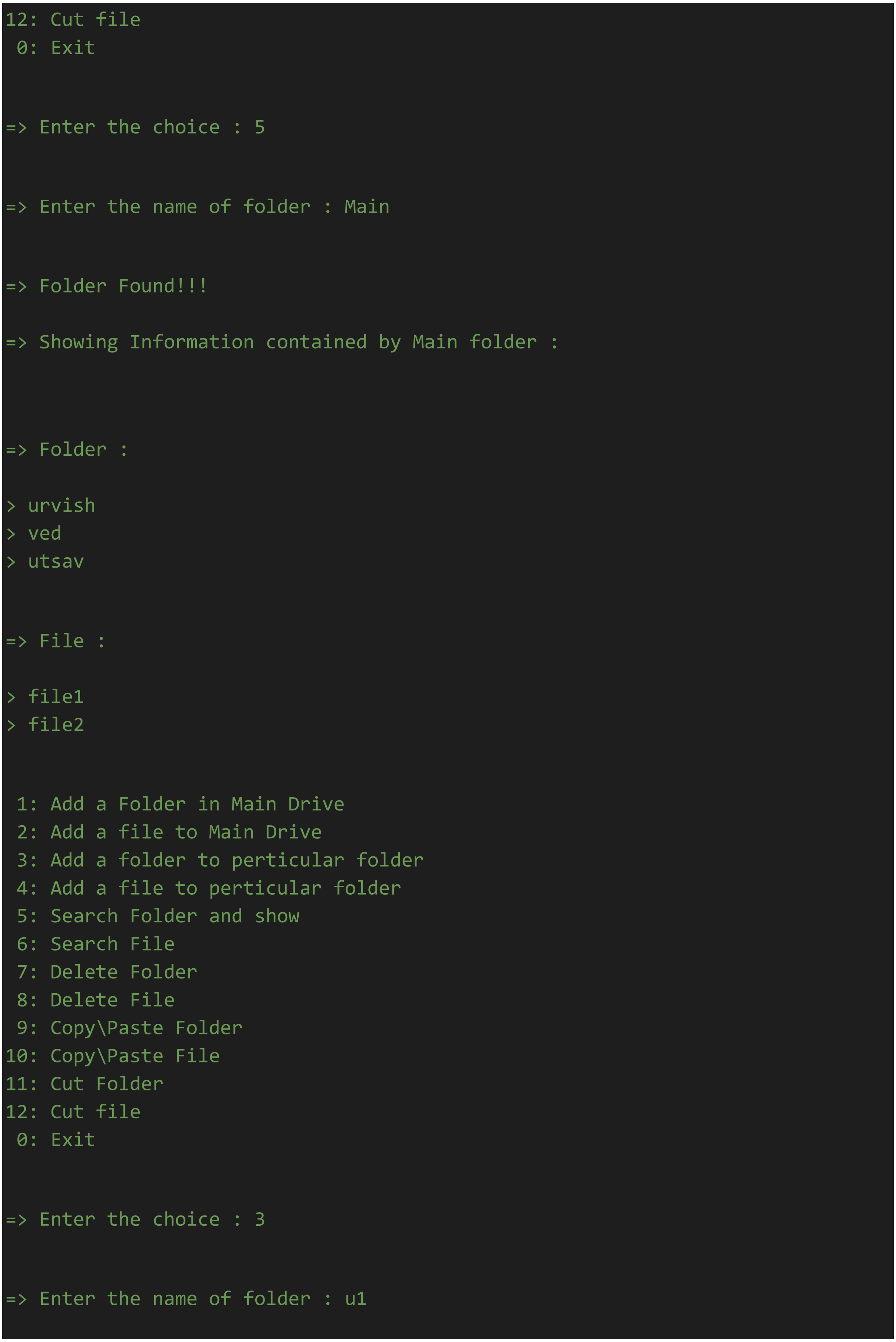
```
/*  Output :



### File Manager [File Management System] ###




 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 1


=> Enter the name of folder : urvish


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
```



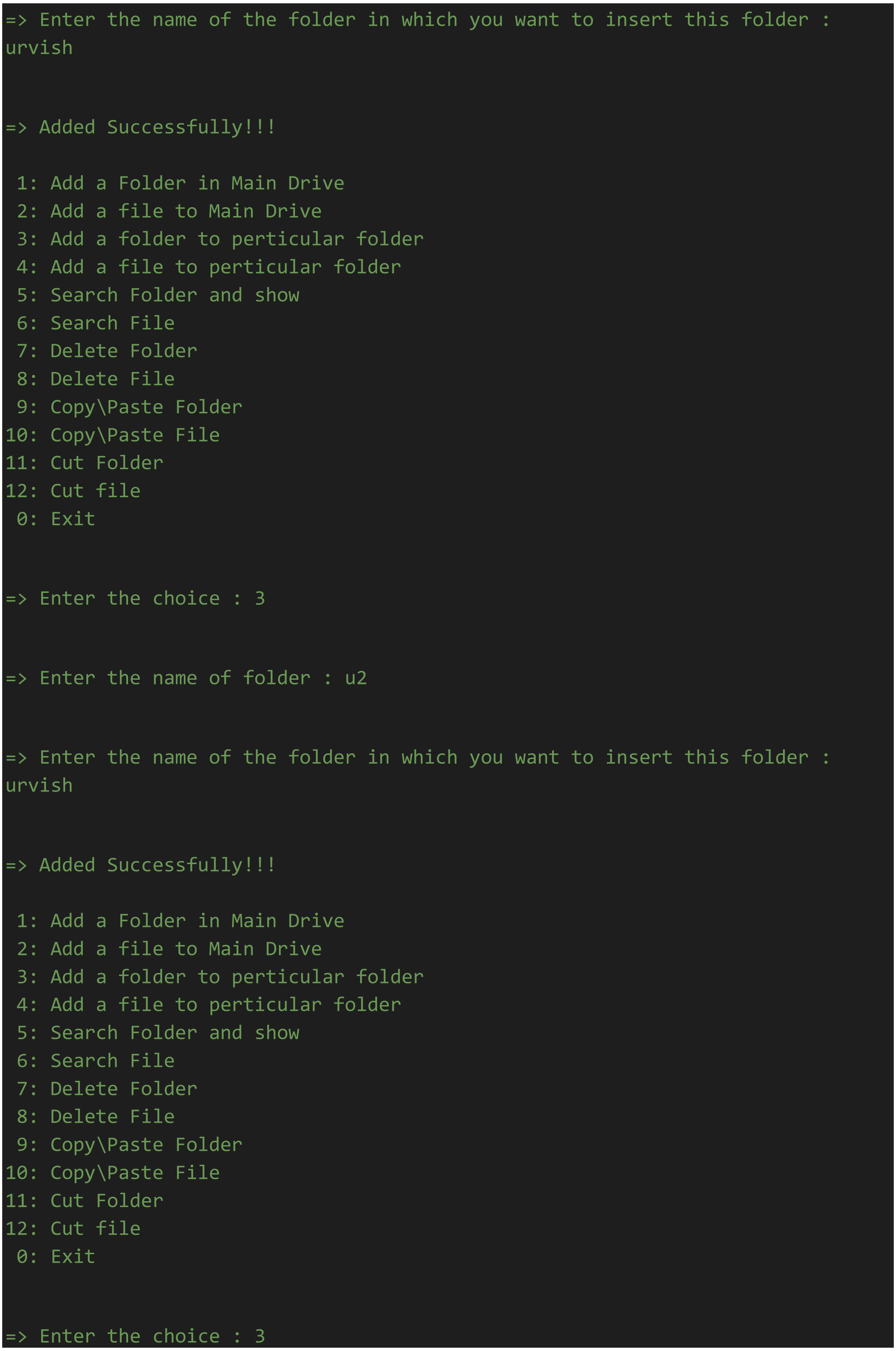
```
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 1


=> Enter the name of folder : ved


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 1


=> Enter the name of folder : utsav


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
```



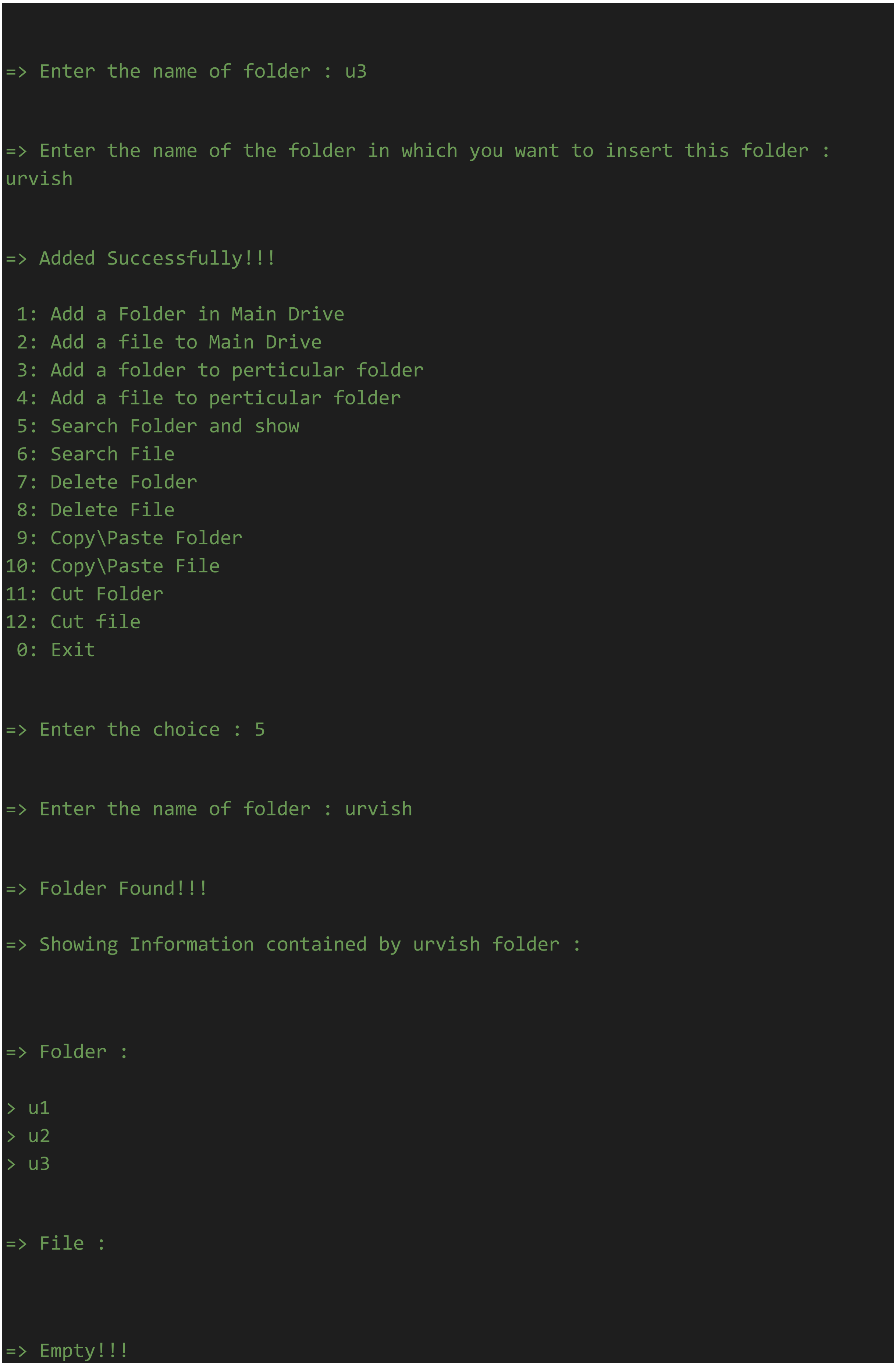
```
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 2


=> Enter the name of file : file1


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 2


=> Enter the name of file : file2


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
```



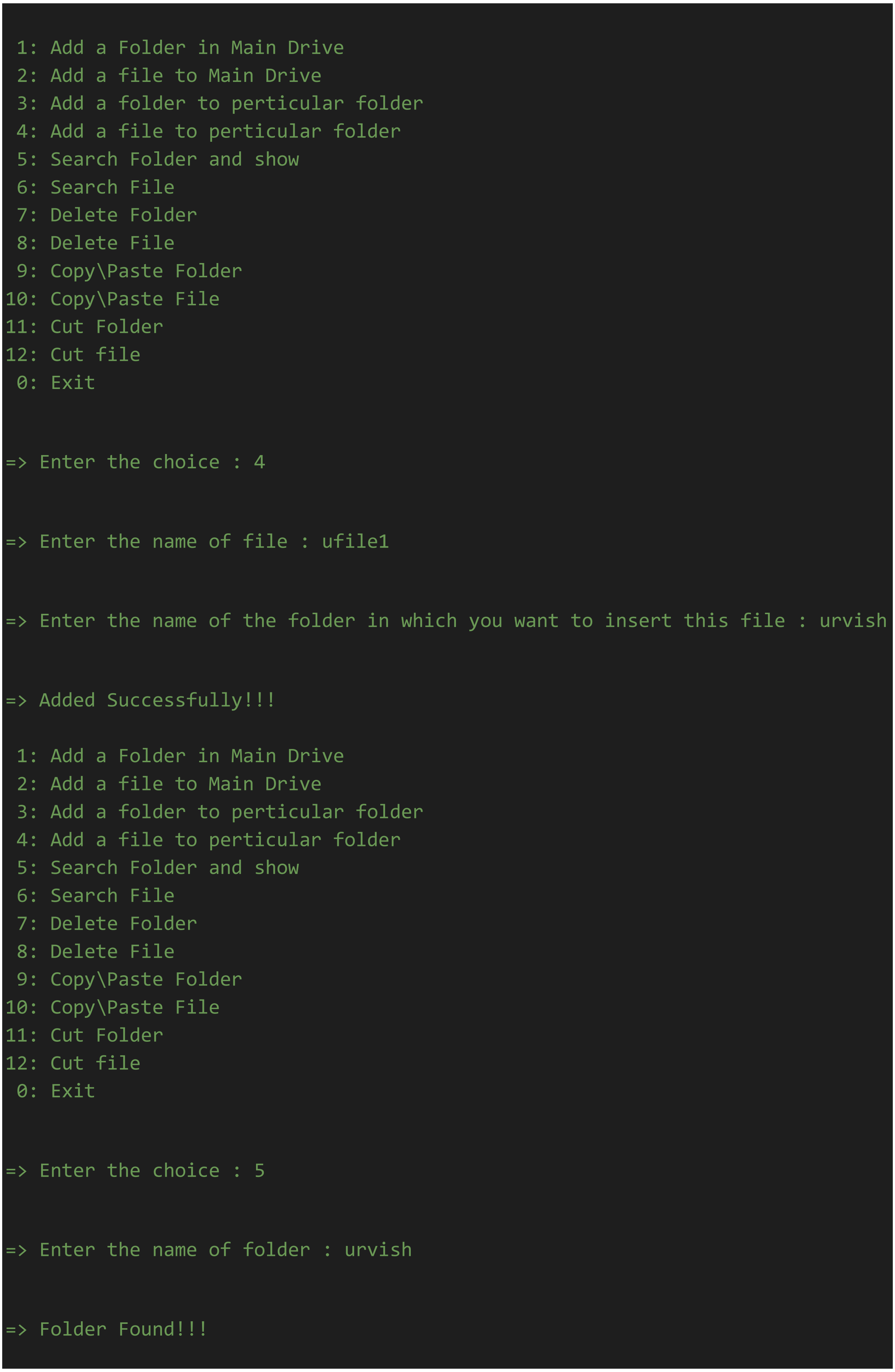
```
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : Main


=> Folder Found!!!

=> Showing Information contained by Main folder :



=> Folder :

> urvish
> ved
> utsav


=> File :

> file1
> file2


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 3


=> Enter the name of folder : u1
```



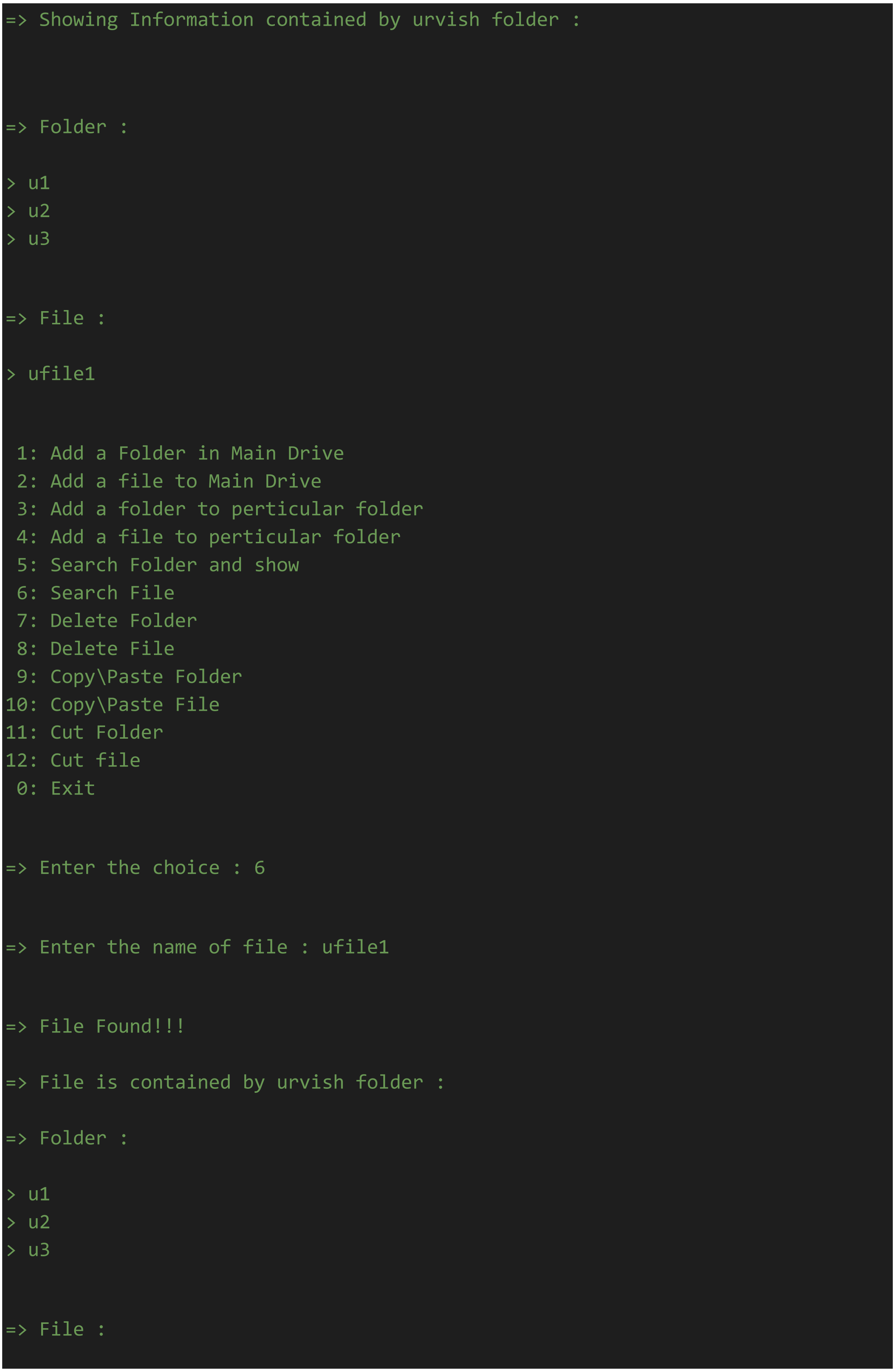
```
=> Enter the name of the folder in which you want to insert this folder :
urvish


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 3


=> Enter the name of folder : u2


=> Enter the name of the folder in which you want to insert this folder :
urvish


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 3
```



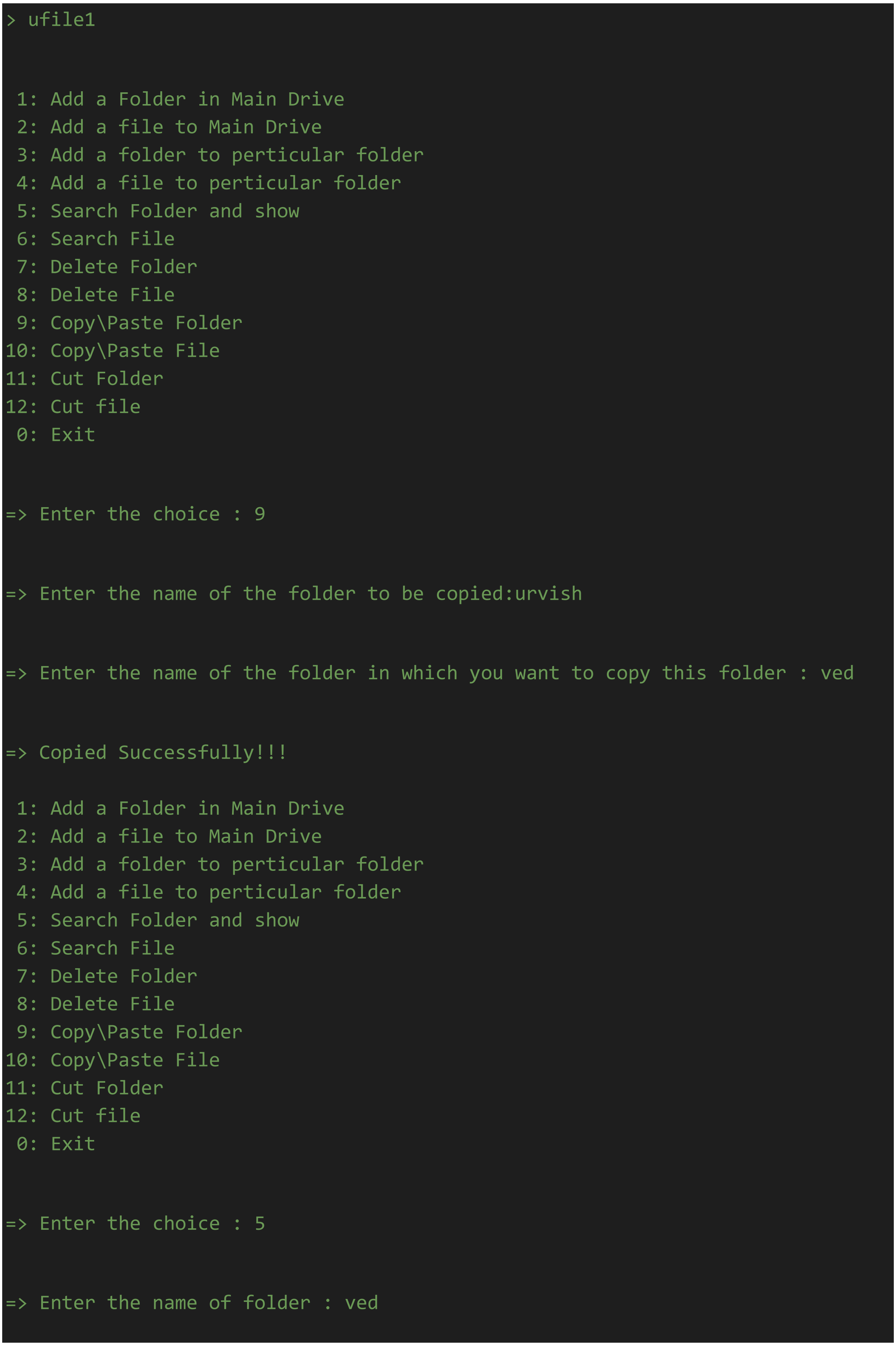
```
=> Enter the name of folder : u3


=> Enter the name of the folder in which you want to insert this folder :
urvish


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : urvish


=> Folder Found!!!

=> Showing Information contained by urvish folder :



=> Folder :

> u1
> u2
> u3


=> File :



=> Empty!!!
```



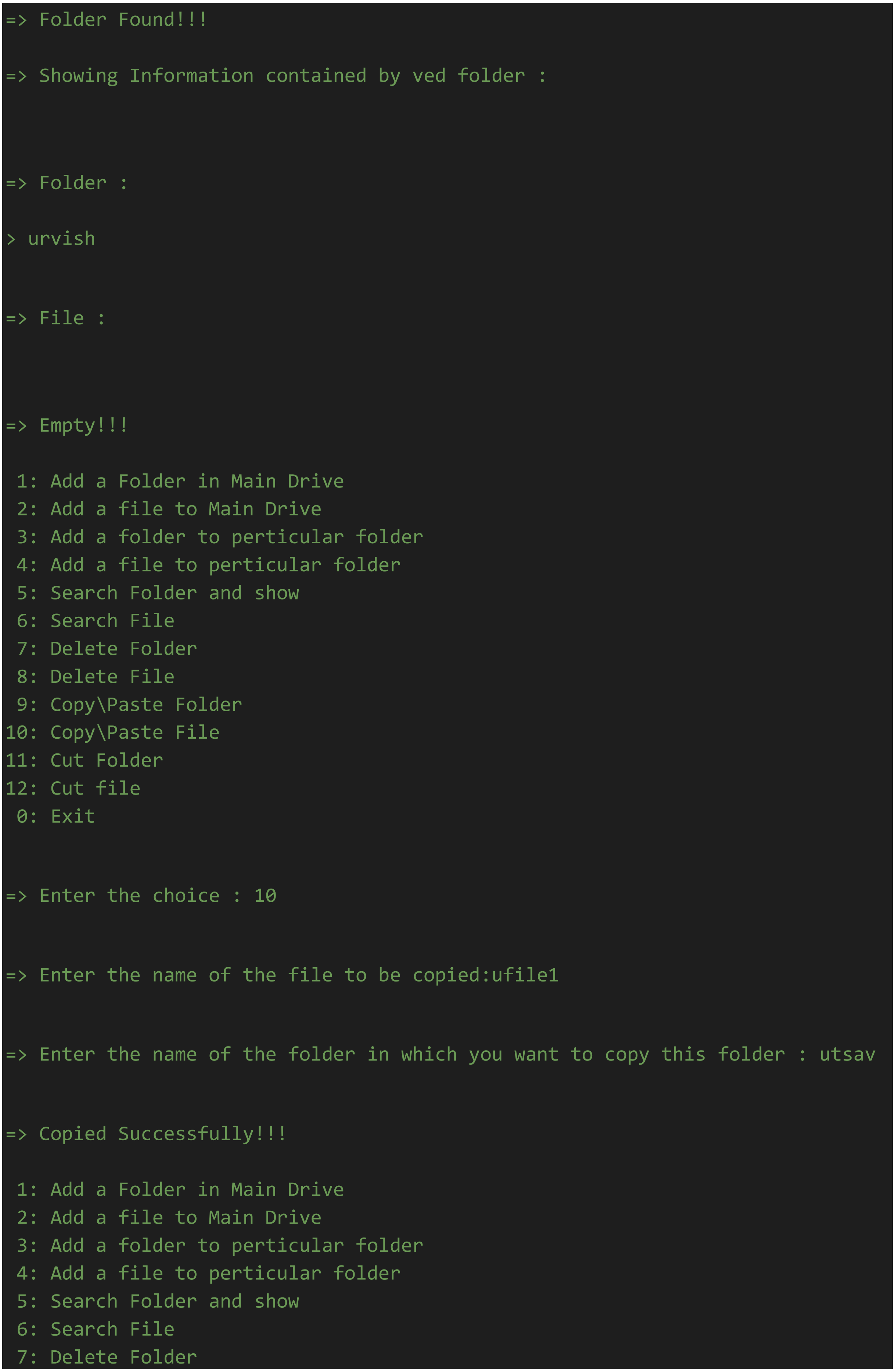
```
 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 4


=> Enter the name of file : ufile1


=> Enter the name of the folder in which you want to insert this file : urvish


=> Added Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : urvish


=> Folder Found!!!
```



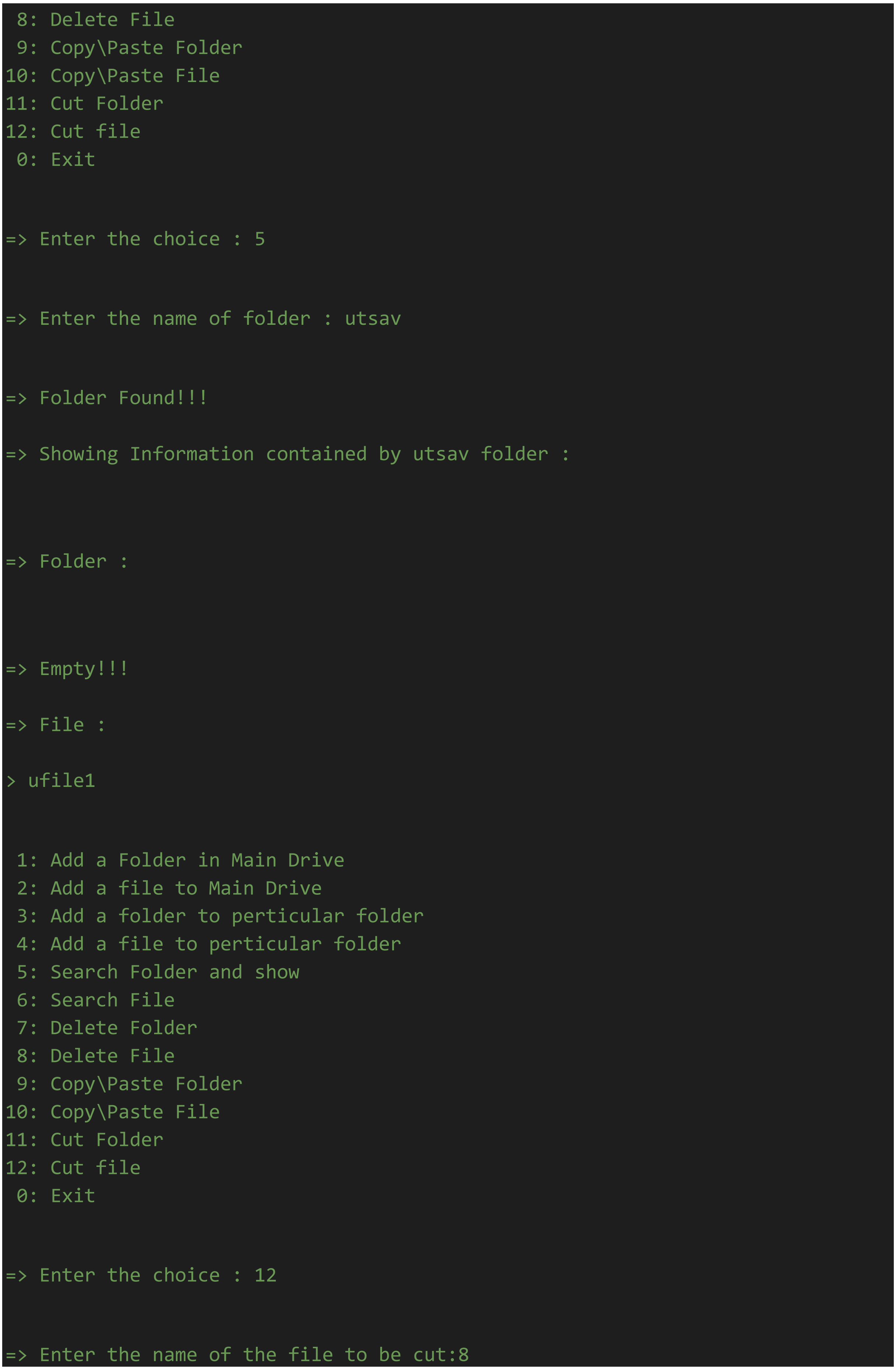
```
=> Showing Information contained by urvish folder :



=> Folder :

> u1
> u2
> u3


=> File :

> ufile1


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 6


=> Enter the name of file : ufile1


=> File Found!!!

=> File is contained by urvish folder :

=> Folder :

> u1
> u2
> u3


=> File :
```



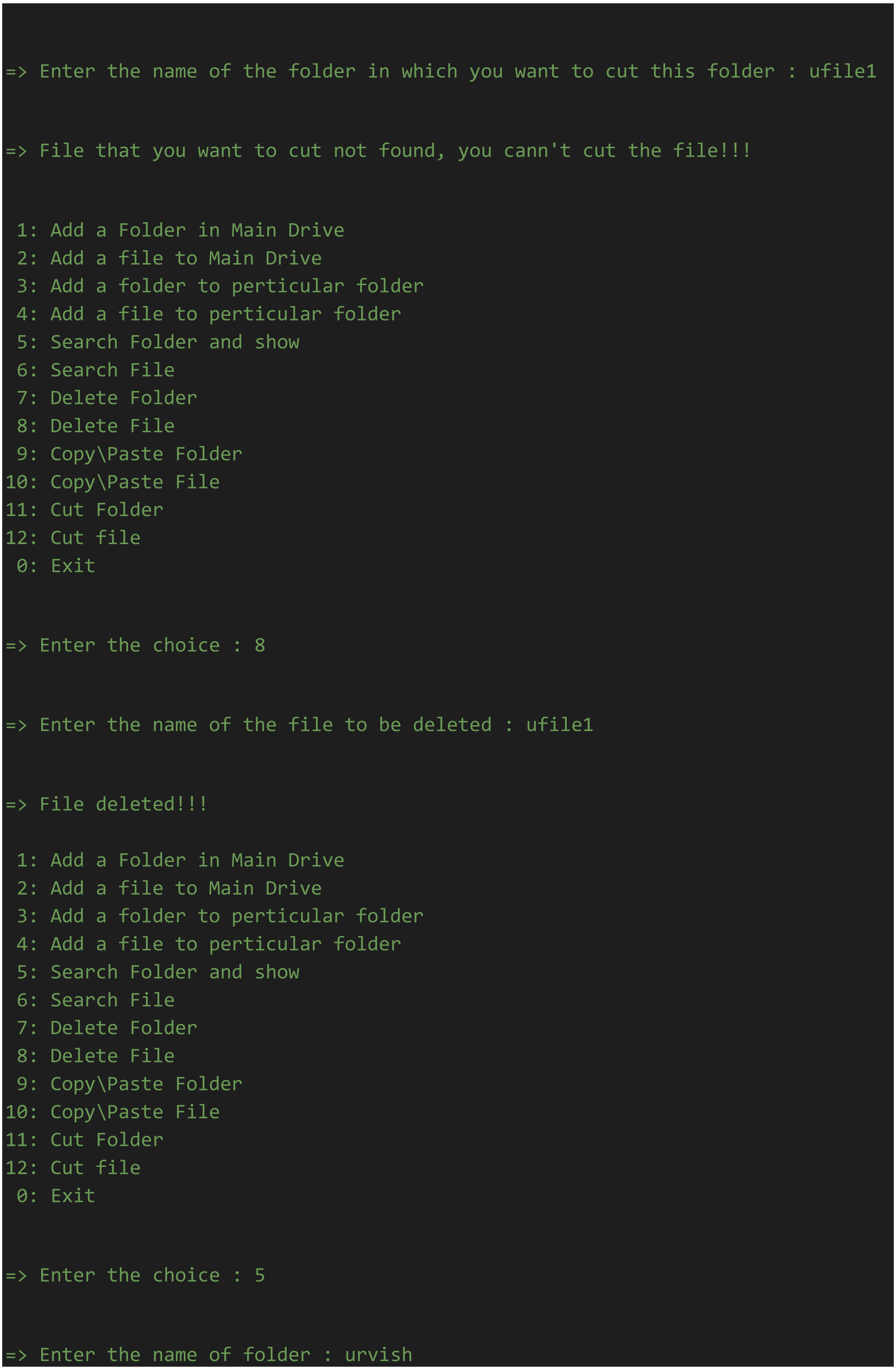
```
> ufile1


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 9


=> Enter the name of the folder to be copied:urvish


=> Enter the name of the folder in which you want to copy this folder : ved


=> Copied Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : ved
```



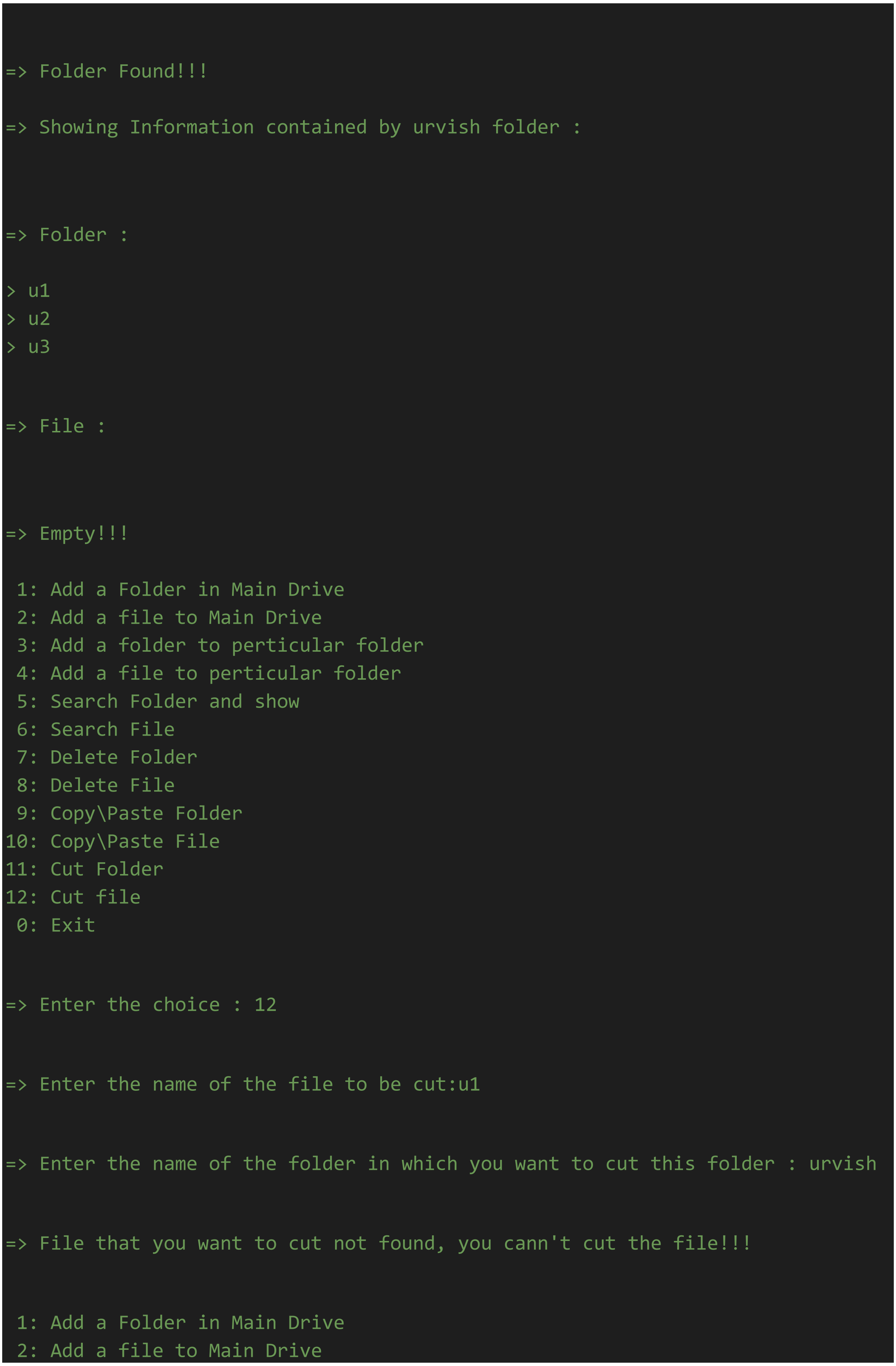
```
=> Folder Found!!!

=> Showing Information contained by ved folder :



=> Folder :

> urvish


=> File :



=> Empty!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 10


=> Enter the name of the file to be copied:ufile1


=> Enter the name of the folder in which you want to copy this folder : utsav


=> Copied Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
```



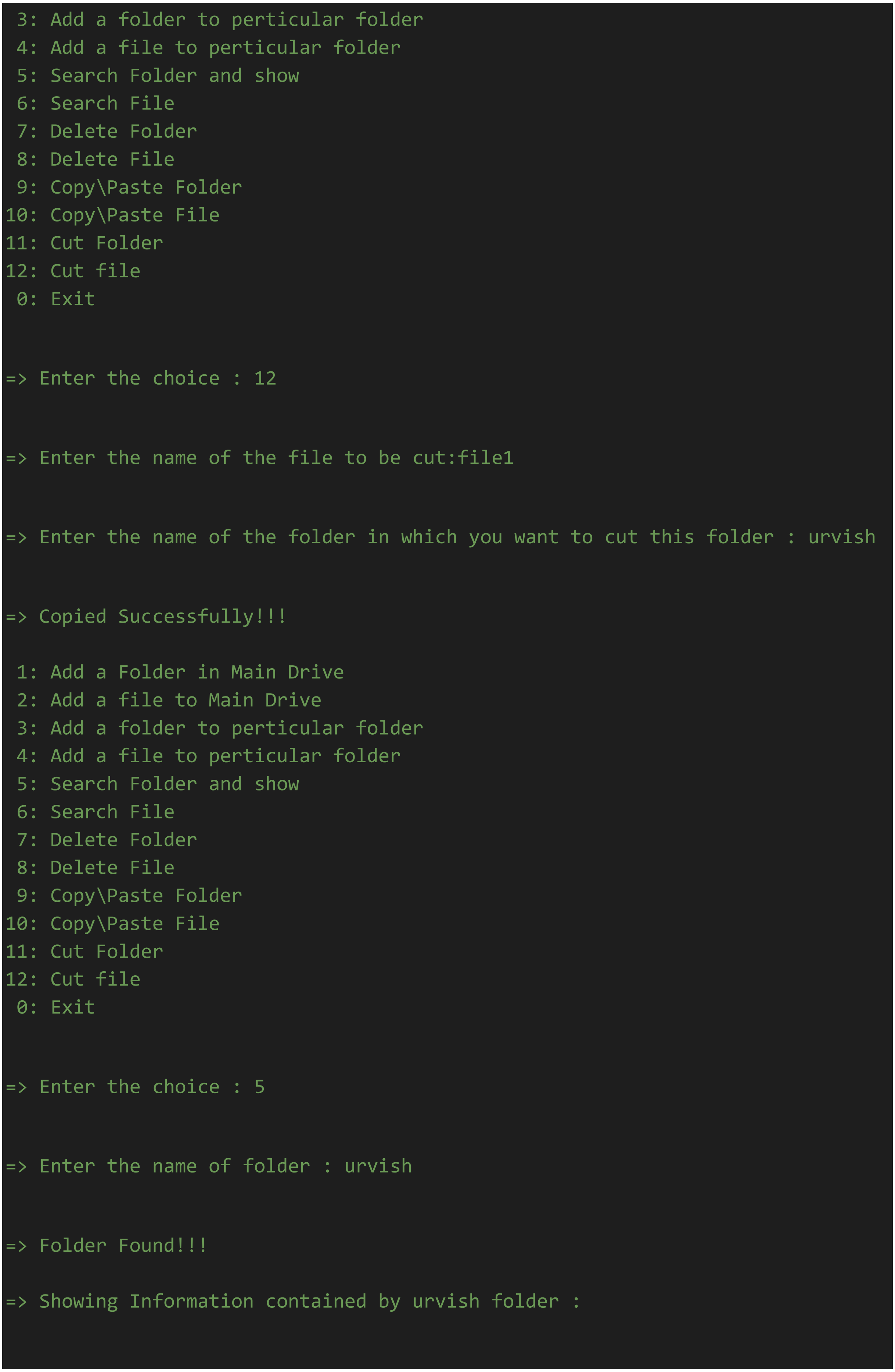
```
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : utsav


=> Folder Found!!!

=> Showing Information contained by utsav folder :



=> Folder :



=> Empty!!!

=> File :

> ufile1


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 12


=> Enter the name of the file to be cut:8
```



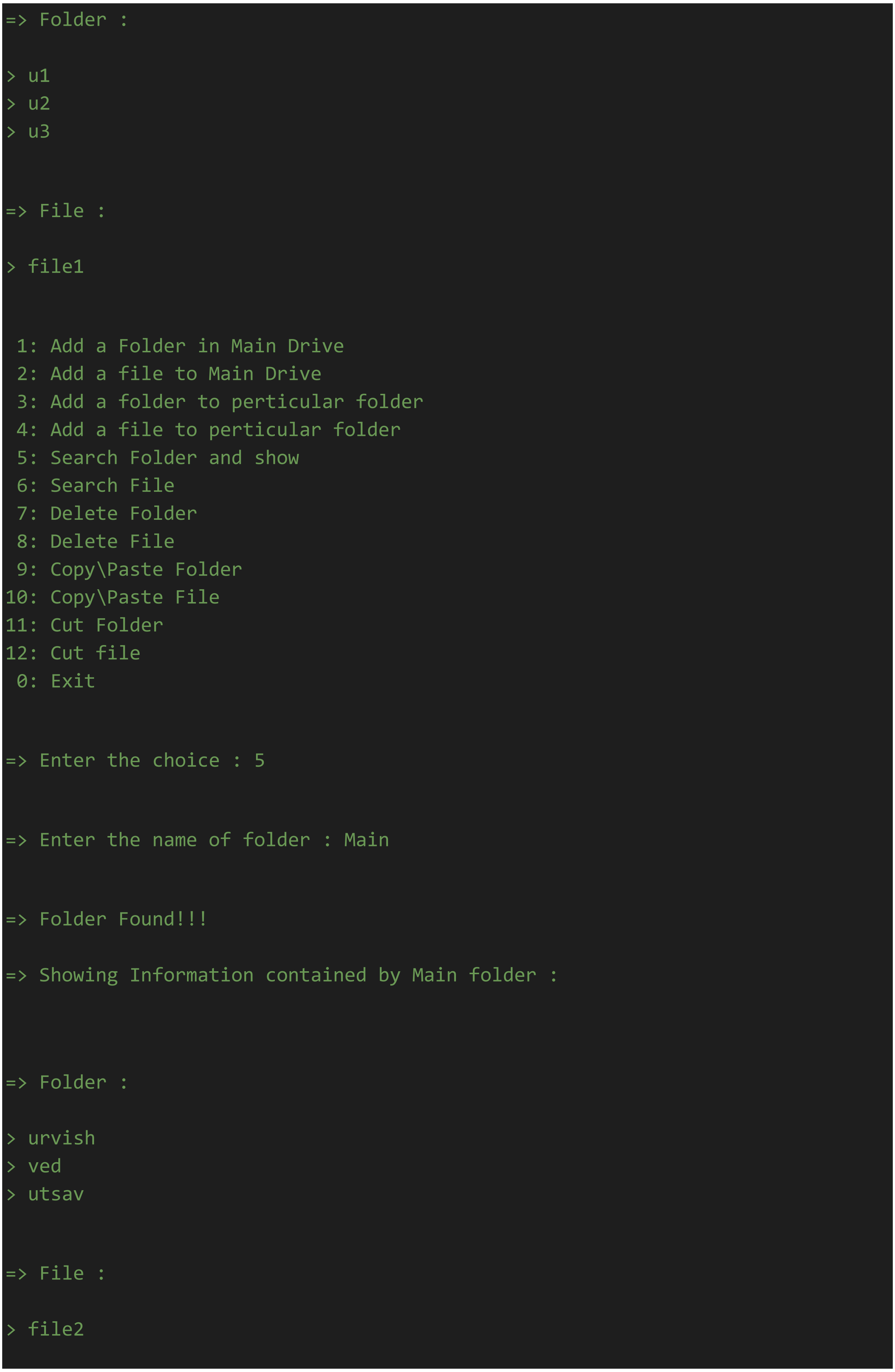
```
=> Enter the name of the folder in which you want to cut this folder : ufile1


=> File that you want to cut not found, you cann't cut the file!!!


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 8


=> Enter the name of the file to be deleted : ufile1


=> File deleted!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : urvish
```



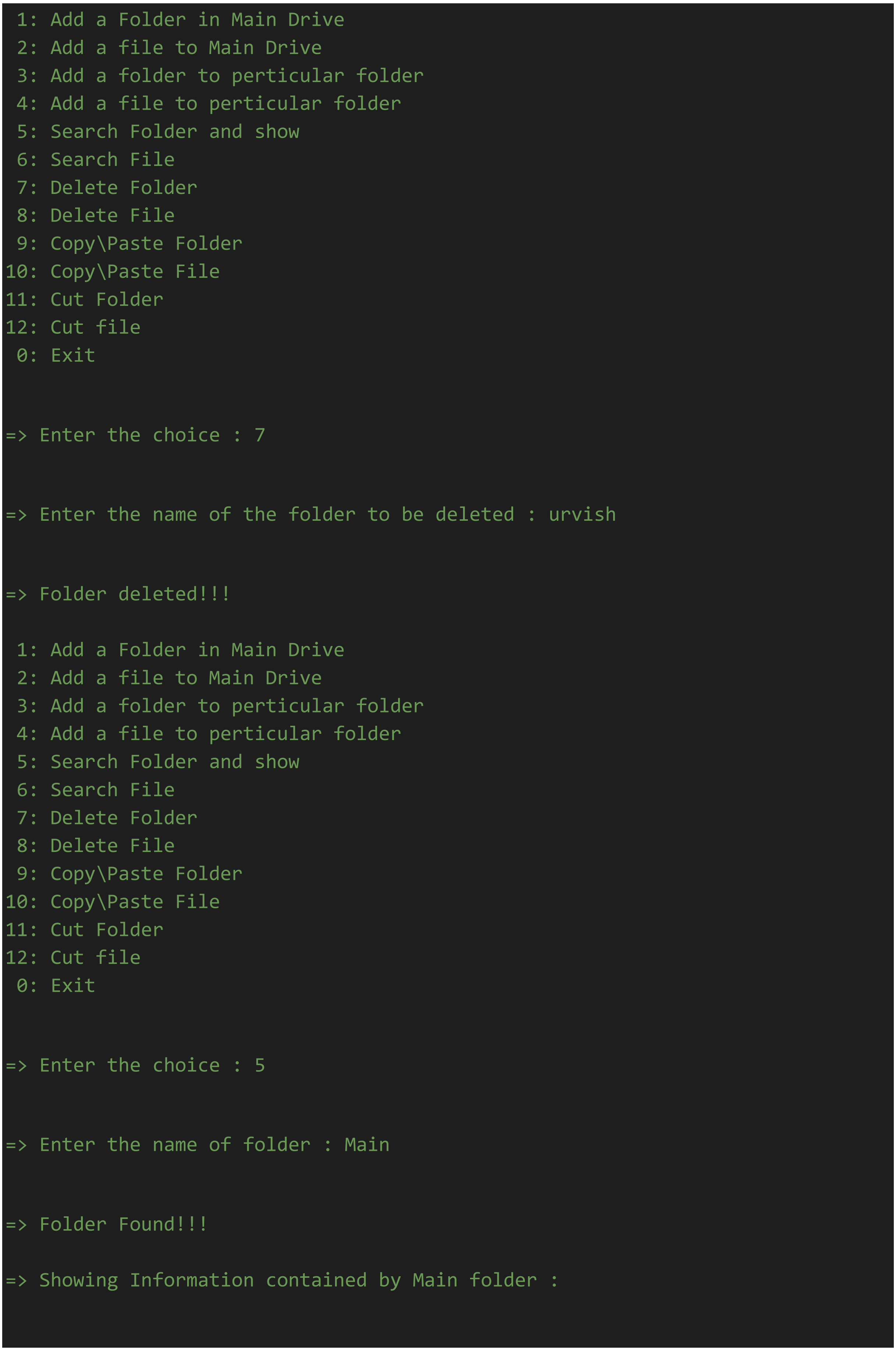
```
=> Folder Found!!!

=> Showing Information contained by urvish folder :



=> Folder :

> u1
> u2
> u3


=> File :



=> Empty!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 12


=> Enter the name of the file to be cut:u1


=> Enter the name of the folder in which you want to cut this folder : urvish


=> File that you want to cut not found, you cann't cut the file!!!


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
```



```
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 12


=> Enter the name of the file to be cut:file1


=> Enter the name of the folder in which you want to cut this folder : urvish


=> Copied Successfully!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : urvish


=> Folder Found!!!

=> Showing Information contained by urvish folder :
```

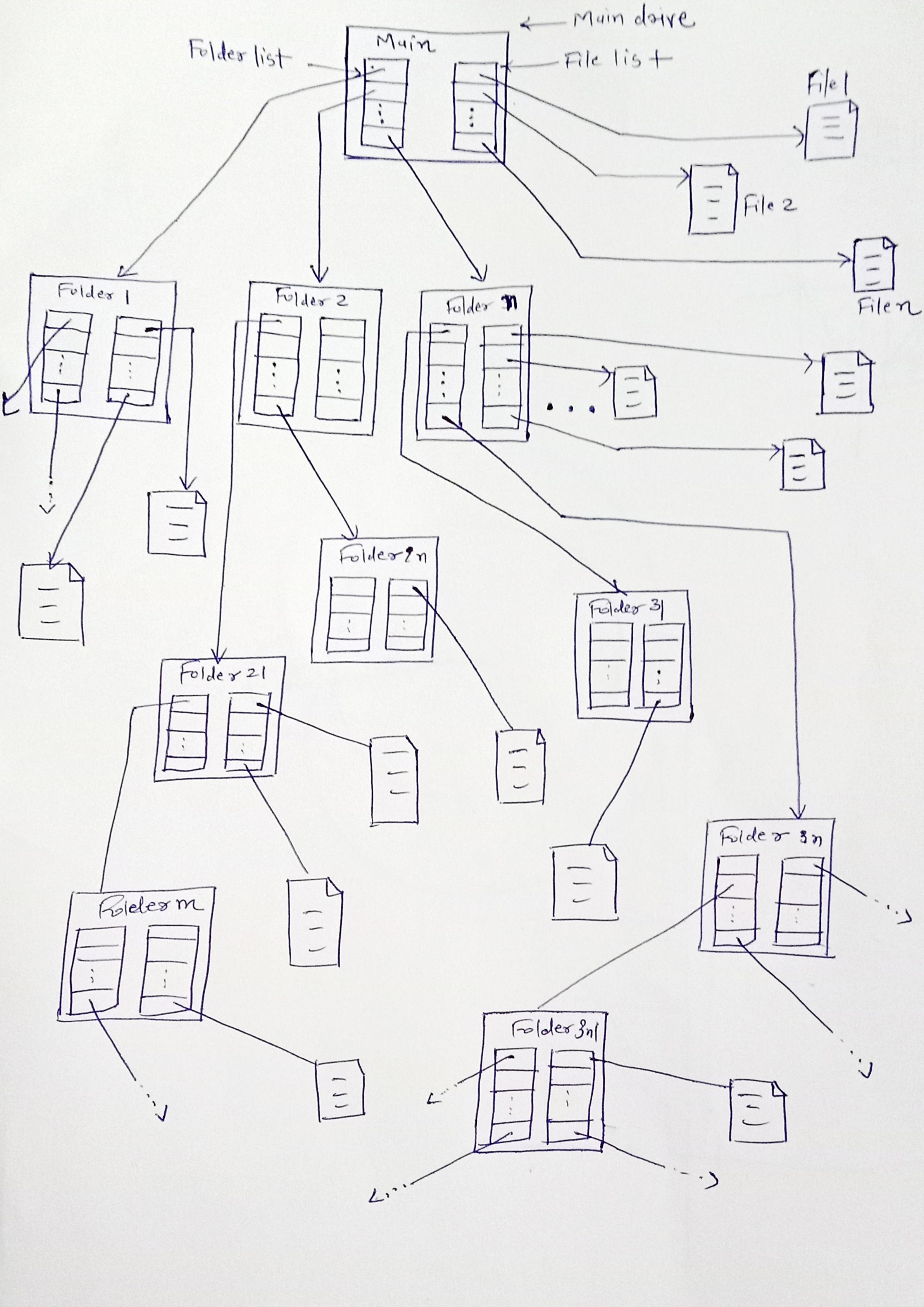


```
=> Folder :

> u1
> u2
> u3


=> File :

> file1


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : Main


=> Folder Found!!!

=> Showing Information contained by Main folder :



=> Folder :

> urvish
> ved
> utsav


=> File :

> file2
```



```
 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 7


=> Enter the name of the folder to be deleted : urvish


=> Folder deleted!!!

 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 5


=> Enter the name of folder : Main


=> Folder Found!!!

=> Showing Information contained by Main folder :
```

```
=> Folder :

> ved
> utsav


=> File :

> file2


 1: Add a Folder in Main Drive
 2: Add a file to Main Drive
 3: Add a folder to perticular folder
 4: Add a file to perticular folder
 5: Search Folder and show
 6: Search File
 7: Delete Folder
 8: Delete File
 9: Copy\Paste Folder
10: Copy\Paste File
11: Cut Folder
12: Cut file
 0: Exit


=> Enter the choice : 0


=> Exit




*/
```


Main doive
Main
Folder list
File list
File1
File 2
Filen
Folder 1
Folder 2
Folder n
Folder 2n
Folder 31
Folder 21
Folder 3n
Folder n
Folder 3n1