DSA Mini-Project

Title: File Management System

Team Members:

Nannapaneni Sashank (PES2UG22CS338)
Nikhil Upadhyay (PES2UG22CS358)
Nishank Bhowal (PES2UG22CS366)
Nishanth J Modpur (PES2UG22CS369)

Synopsis

The file management system is designed to simplify the management of files for users. It offers a straightforward and intuitive interface, presenting a range of options to manage files efficiently.

Key Features:

1. Creating a File: Users can effortlessly create new files with just a few clicks. The system provides a user-friendly interface for specifying the file name.
2. Deleting a File: With the File Management System, users can easily delete files they no longer need. This application offers a secure and straightforward deletion process to prevent accidental file loss.
3. Searching for a File: Need to find a specific file quickly? The system's search feature allows users to locate files based on file names, ensuring a fast and efficient search experience.
4. Restoring Deleted Files: The system includes a "Restore" option, allowing users to recover deleted files.
5. Displaying All Files in Directory: Users can view a comprehensive list of all their files, making it easy to manage and organize their files. The system provides a well-structured and organized display of file information.

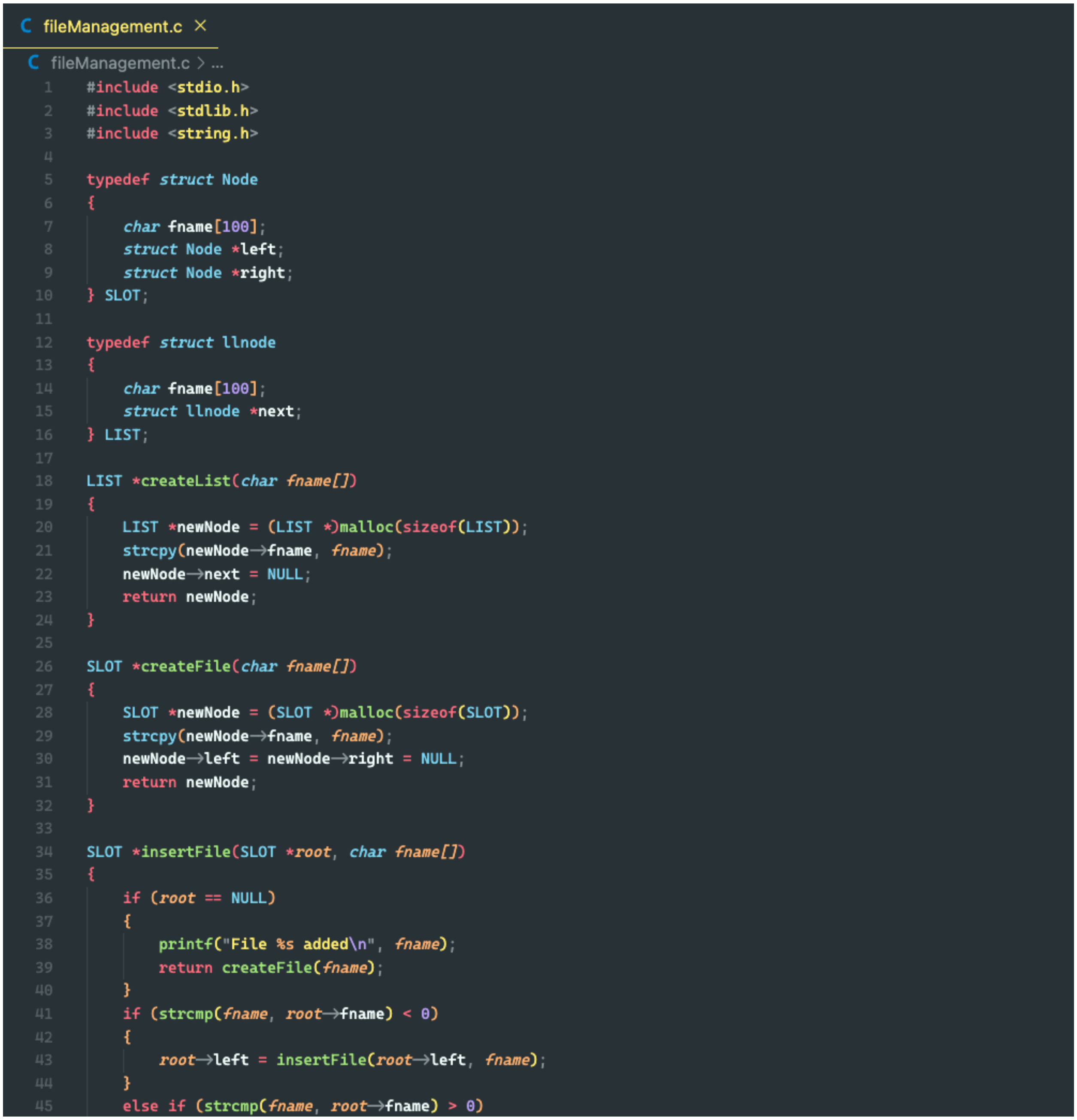
6. View Files with Certain Extension: Users can obtain a list of files ending with a certain extension to optimize their search experience by simply entering the extension as input.

ADT Definition of Data Structures Used

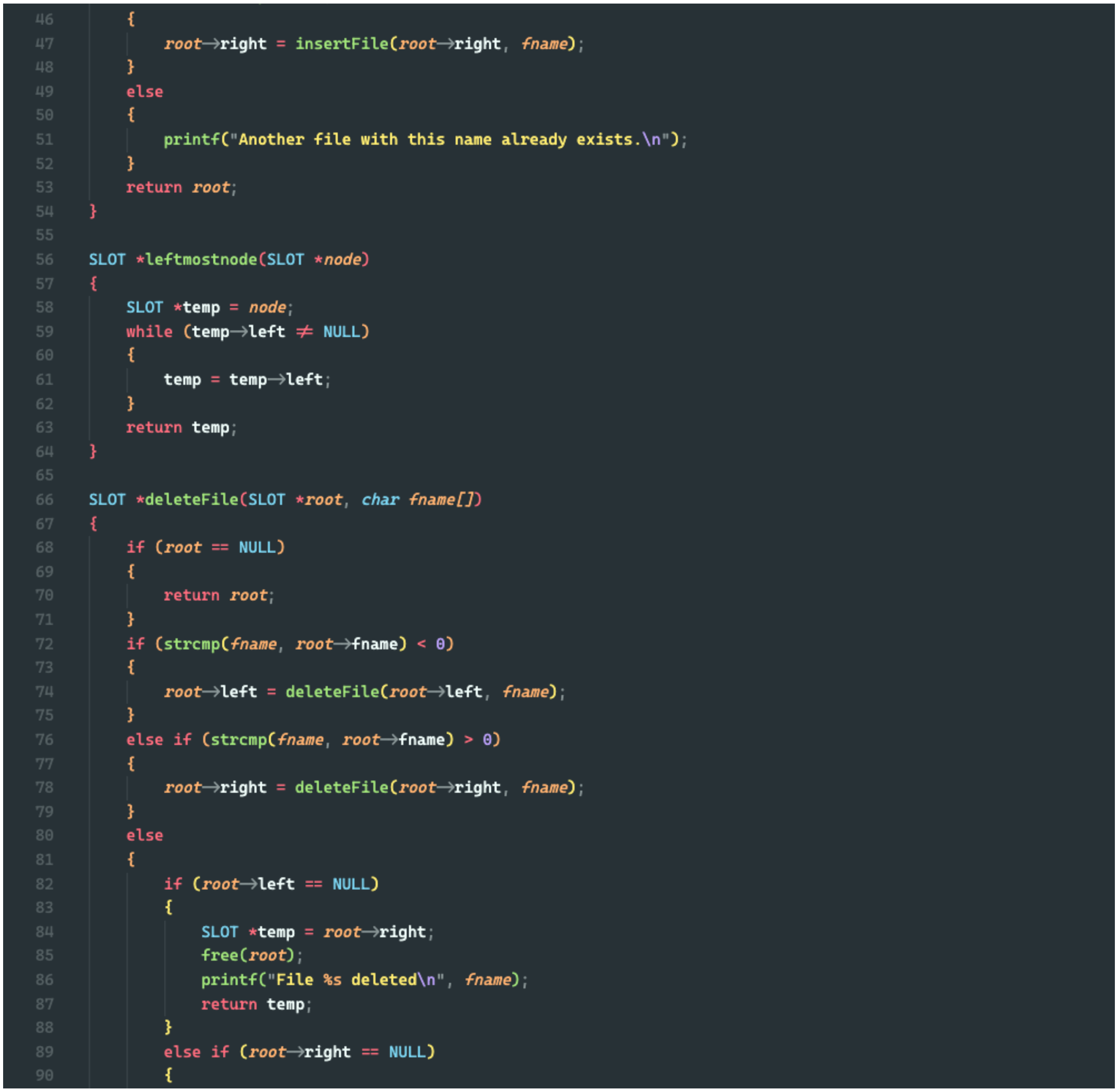
- SLOT *createFile(char fname[]):
  Creates a new file node with the given name and inserts it into the directory tree.
- SLOT *insertFile(SLOT *root, char fname[]):
  Inserts a new file node into the directory tree while maintaining the binary search tree property.
- SLOT *deleteFile(SLOT *root, char fname[]):
  Deletes a file node with the given name from the directory tree.
- void inOrderTraversal(SLOT *root):
  Performs an in-order traversal of the directory tree to display all files and directories.
- SLOT *searchFile(SLOT *root, char fname[]):
  Searches for a file with the given name in the directory tree.
- void displayFilesWithExtension(SLOT *root, char extension[]):
  Displays files with a specific extension in the directory tree.
- SLOT *leftmostnode(SLOT *node):
  Finds the leftmost node in the directory tree.
  (Returns the least value node on the right of the current node.)
- LIST *createList(char fname[]):
  Creates a new node to store the name of a deleted file and inserts it into the list.
- LIST *insertToList(LIST *head, char fname[]):
  Inserts a new node at the end of the list to record a deleted file.
- LIST *restoreDeletedFile(LIST *head, char fname[]):
  Searches for a deleted file in the list and returns it if found, allowing restoration.

Code With Output Screenshots

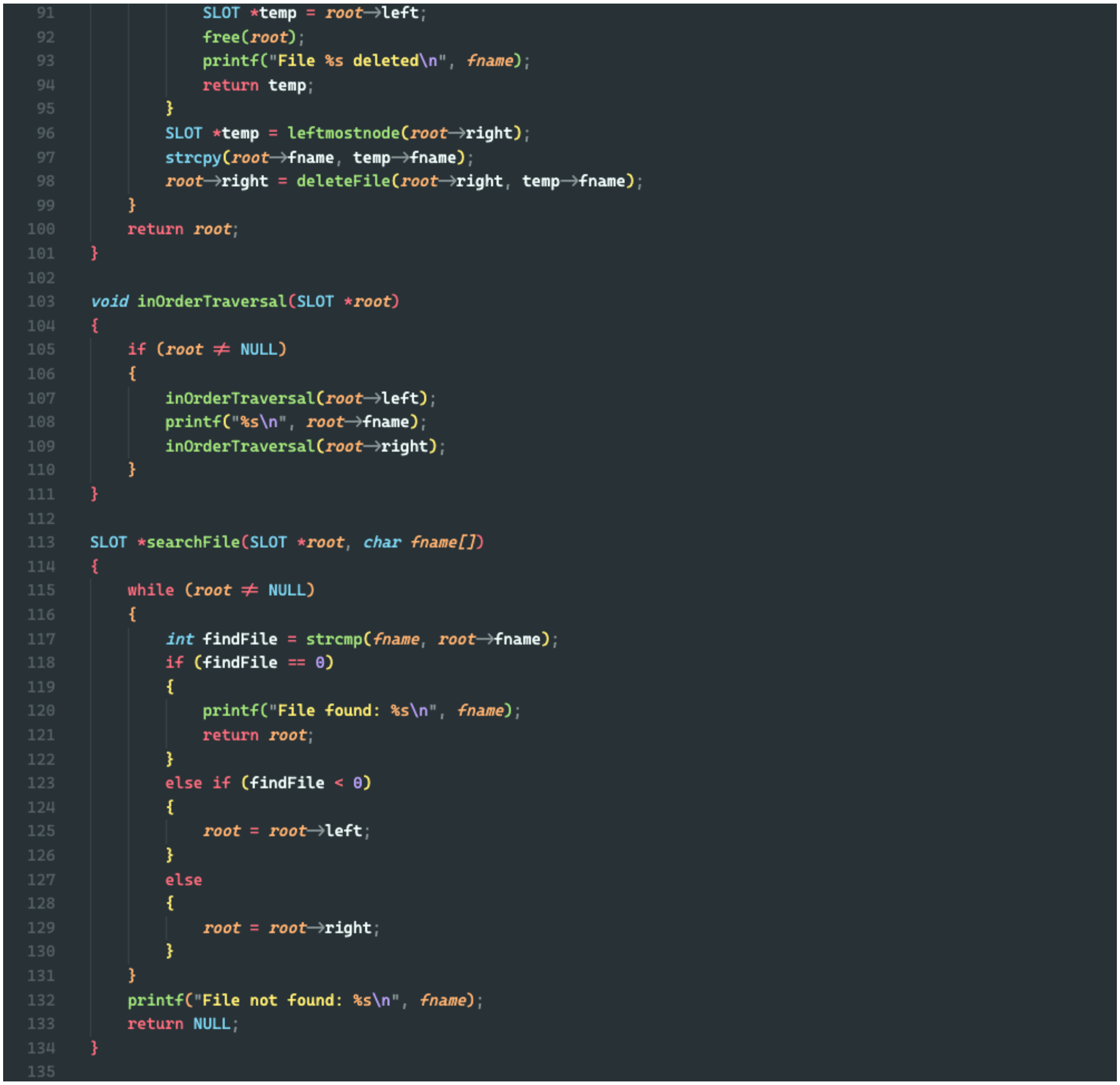
Code



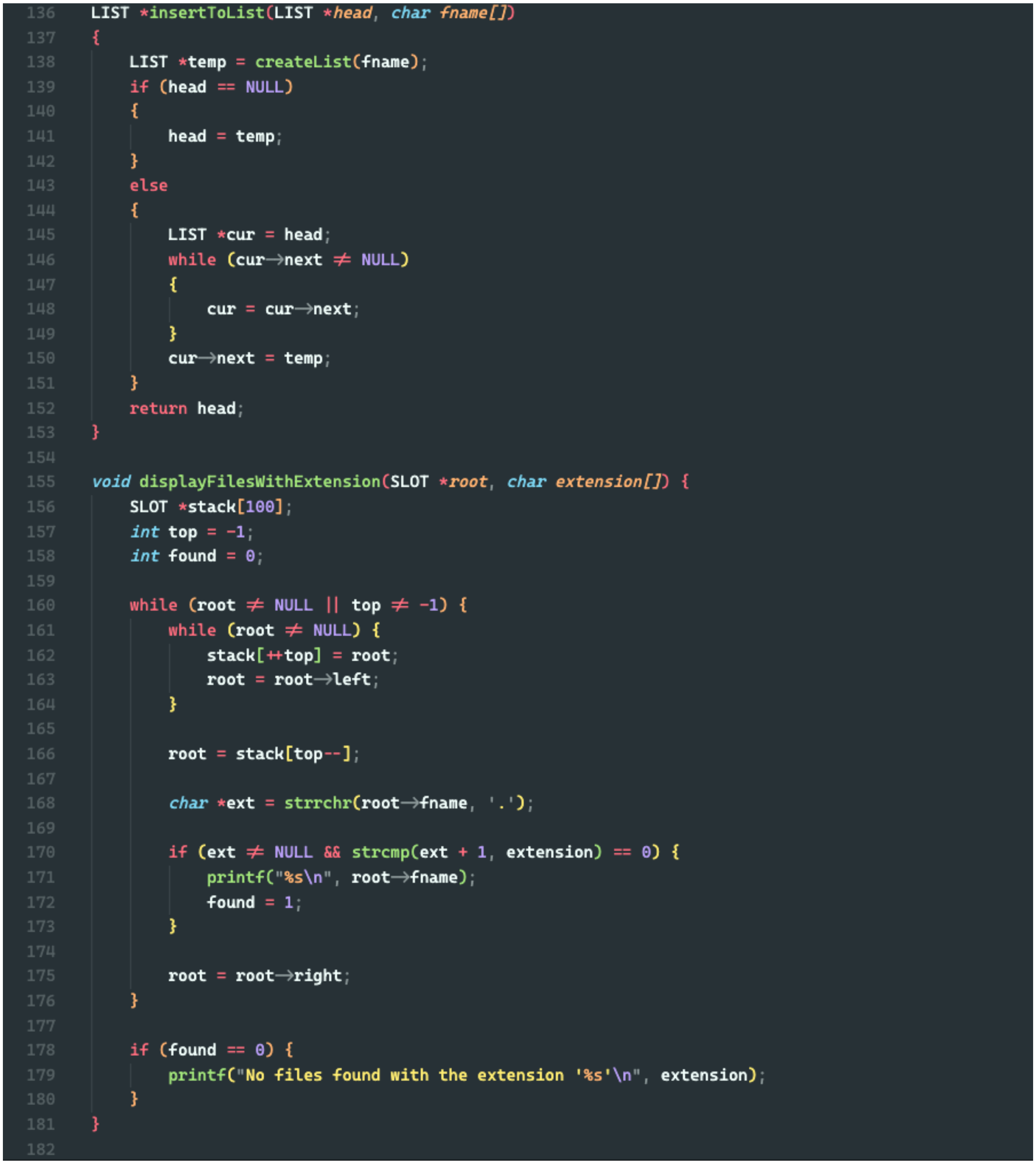
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Node
{
    char fname[100];
    struct Node *left;
    struct Node *right;
} SLOT;

typedef struct llnode
{
    char fname[100];
    struct llnode *next;
} LIST;

LIST *createList(char fname[])
{
    LIST *newNode = (LIST *)malloc(sizeof(LIST));
    strcpy(newNode->fname, fname);
    newNode->next = NULL;
    return newNode;
}

SLOT *createFile(char fname[])
{
    SLOT *newNode = (SLOT *)malloc(sizeof(SLOT));
    strcpy(newNode->fname, fname);
    newNode->left = newNode->right = NULL;
    return newNode;
}

SLOT *insertFile(SLOT *root, char fname[])
{
    if (root == NULL)
    {
        printf("File %s added\n", fname);
        return createFile(fname);
    }
    if (strcmp(fname, root->fname) < 0)
    {
        root->left = insertFile(root->left, fname);
    }
    else if (strcmp(fname, root->fname) > 0)
```



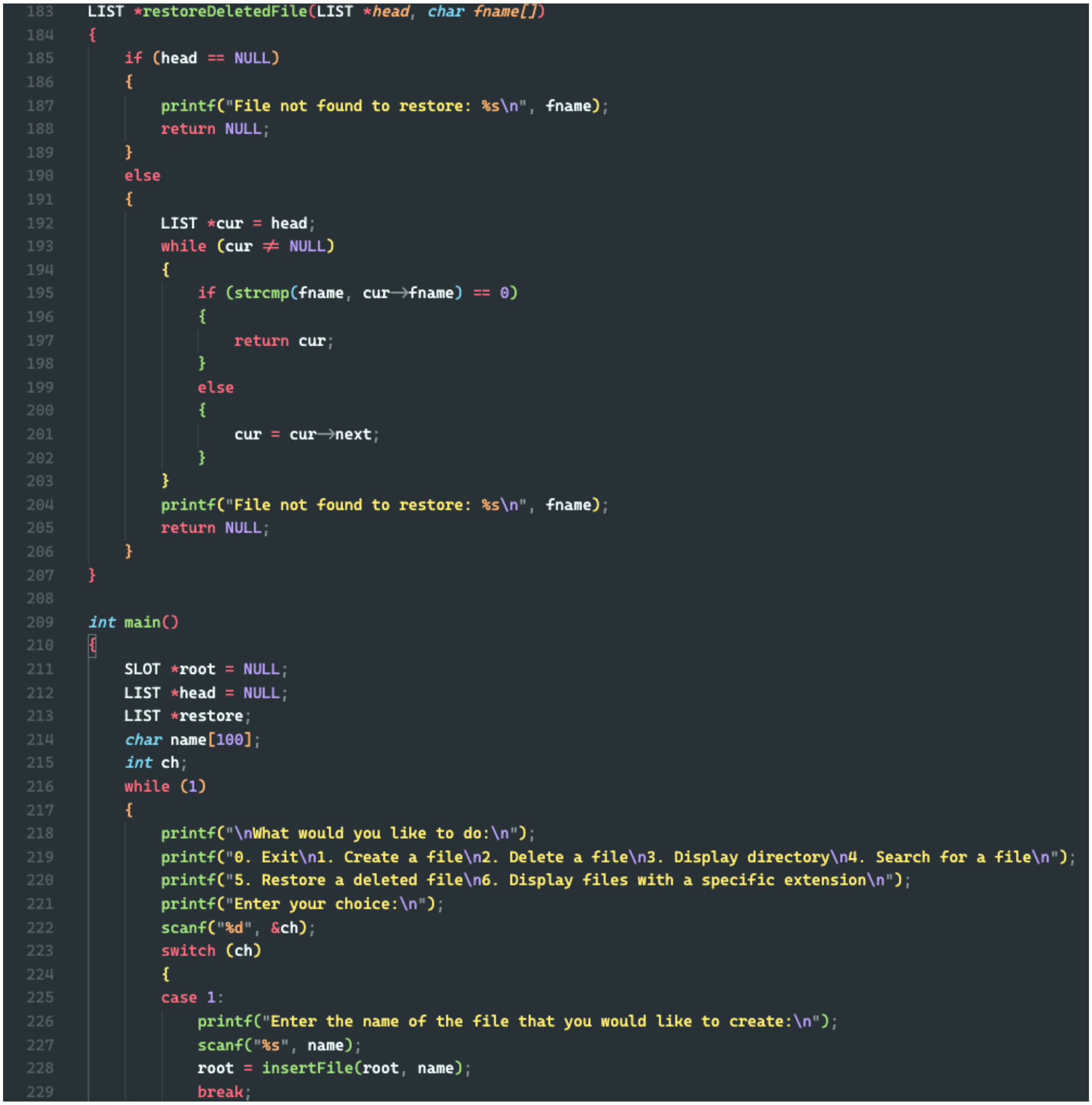
```c
    {
        root->right = insertFile(root->right, fname);
    }
    else
    {
        printf("Another file with this name already exists.\n");
    }
    return root;
}

SLOT *leftmostnode(SLOT *node)
{
    SLOT *temp = node;
    while (temp->left != NULL)
    {
        temp = temp->left;
    }
    return temp;
}

SLOT *deleteFile(SLOT *root, char fname[])
{
    if (root == NULL)
    {
        return root;
    }
    if (strcmp(fname, root->fname) < 0)
    {
        root->left = deleteFile(root->left, fname);
    }
    else if (strcmp(fname, root->fname) > 0)
    {
        root->right = deleteFile(root->right, fname);
    }
    else
    {
        if (root->left == NULL)
        {
            SLOT *temp = root->right;
            free(root);
            printf("File %s deleted\n", fname);
            return temp;
        }
        else if (root->right == NULL)
        {
```



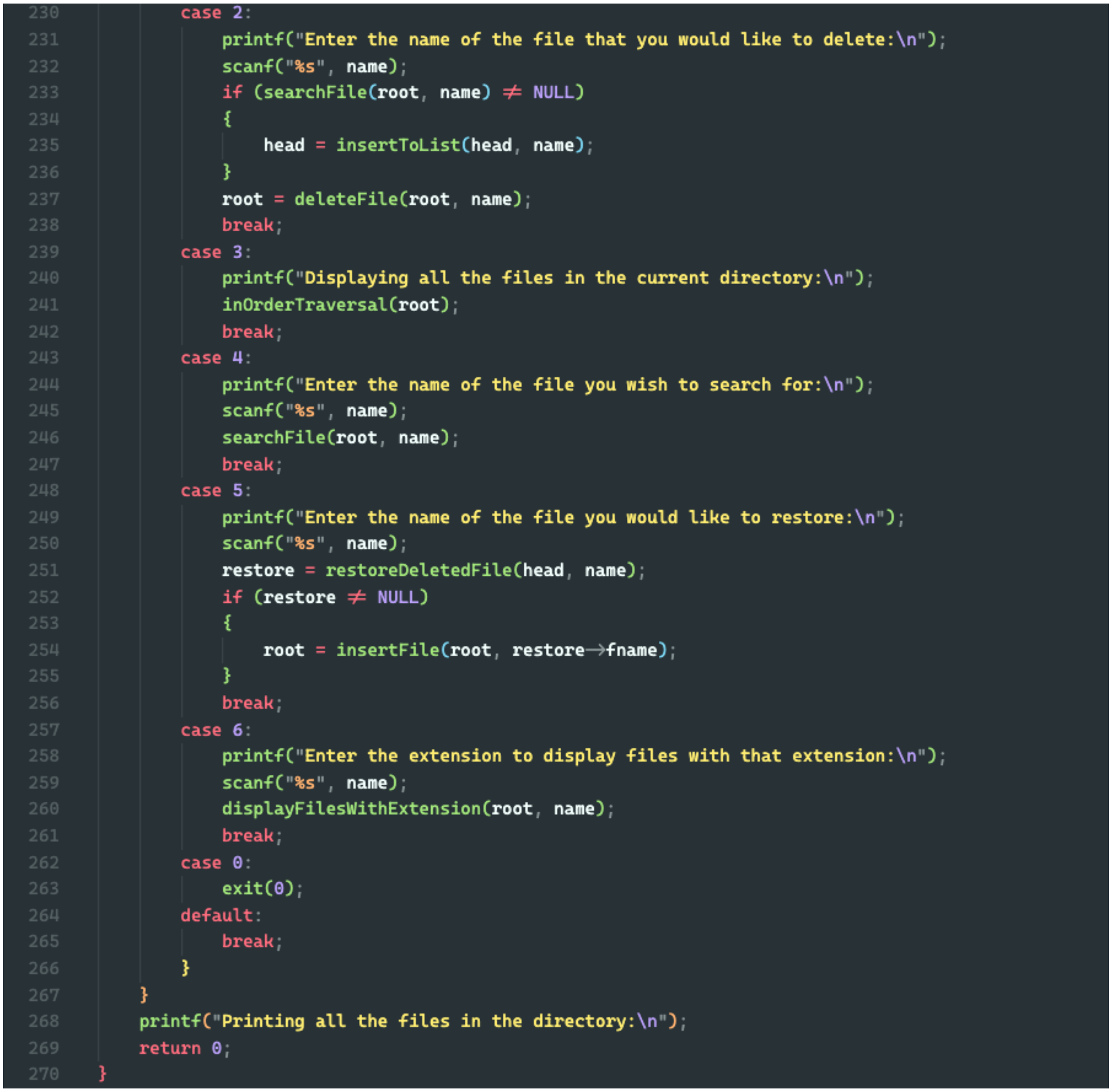
```c
            SLOT *temp = root->left;
            free(root);
            printf("File %s deleted\n", fname);
            return temp;
        }
        SLOT *temp = leftmostnode(root->right);
        strcpy(root->fname, temp->fname);
        root->right = deleteFile(root->right, temp->fname);
    }
    return root;
}

void inOrderTraversal(SLOT *root)
{
    if (root != NULL)
    {
        inOrderTraversal(root->left);
        printf("%s\n", root->fname);
        inOrderTraversal(root->right);
    }
}

SLOT *searchFile(SLOT *root, char fname[])
{
    while (root != NULL)
    {
        int findFile = strcmp(fname, root->fname);
        if (findFile == 0)
        {
            printf("File found: %s\n", fname);
            return root;
        }
        else if (findFile < 0)
        {
            root = root->left;
        }
        else
        {
            root = root->right;
        }
    }
    printf("File not found: %s\n", fname);
    return NULL;
}
```



```c
LIST *insertToList(LIST *head, char fname[])
{
    LIST *temp = createList(fname);
    if (head == NULL)
    {
        head = temp;
    }
    else
    {
        LIST *cur = head;
        while (cur->next != NULL)
        {
            cur = cur->next;
        }
        cur->next = temp;
    }
    return head;
}

void displayFilesWithExtension(SLOT *root, char extension[]) {
    SLOT *stack[100];
    int top = -1;
    int found = 0;

    while (root != NULL || top != -1) {
        while (root != NULL) {
            stack[++top] = root;
            root = root->left;
        }

        root = stack[top--];

        char *ext = strrchr(root->fname, '.');

        if (ext != NULL && strcmp(ext + 1, extension) == 0) {
            printf("%s\n", root->fname);
            found = 1;
        }

        root = root->right;
    }

    if (found == 0) {
        printf("No files found with the extension '%s'\n", extension);
    }
}

```

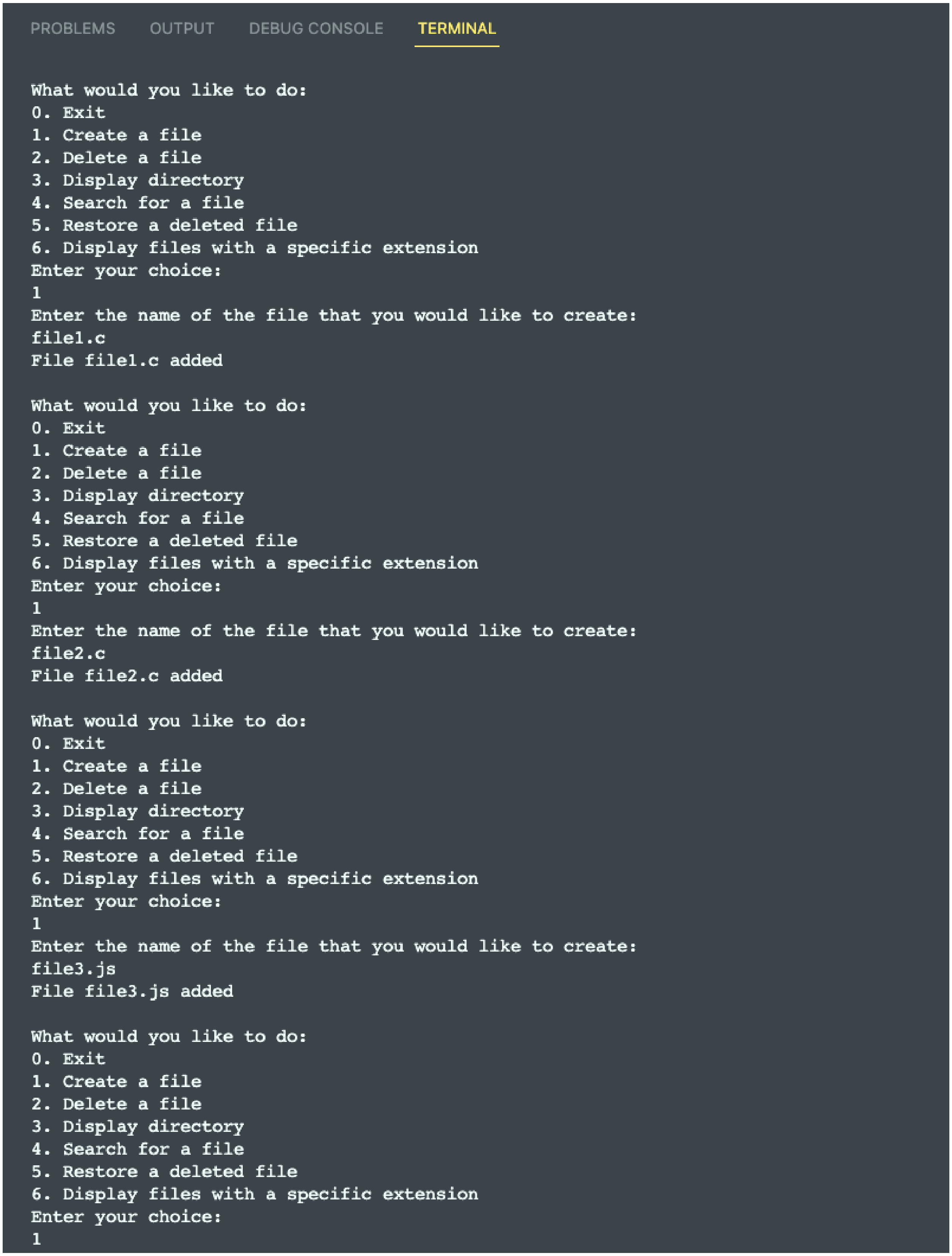


```c
LIST *restoreDeletedFile(LIST *head, char fname[])
{
    if (head == NULL)
    {
        printf("File not found to restore: %s\n", fname);
        return NULL;
    }
    else
    {
        LIST *cur = head;
        while (cur != NULL)
        {
            if (strcmp(fname, cur->fname) == 0)
            {
                return cur;
            }
            else
            {
                cur = cur->next;
            }
        }
        printf("File not found to restore: %s\n", fname);
        return NULL;
    }
}

int main()
{
    SLOT *root = NULL;
    LIST *head = NULL;
    LIST *restore;
    char name[100];
    int ch;
    while (1)
    {
        printf("\nWhat would you like to do:\n");
        printf("0. Exit\n1. Create a file\n2. Delete a file\n3. Display directory\n4. Search for a file\n");
        printf("5. Restore a deleted file\n6. Display files with a specific extension\n");
        printf("Enter your choice:\n");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            printf("Enter the name of the file that you would like to create:\n");
            scanf("%s", name);
            root = insertFile(root, name);
            break;
```

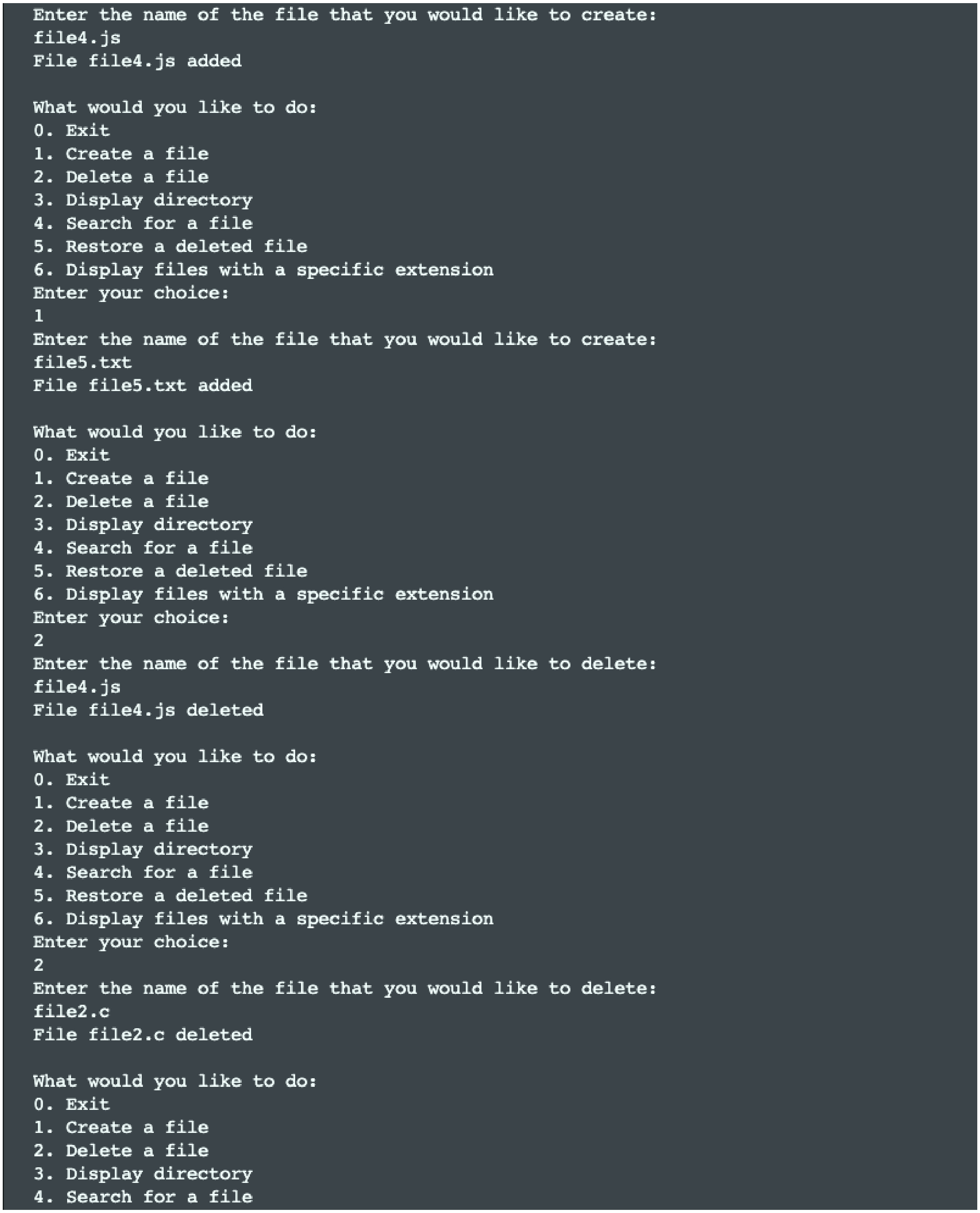


```c
        case 2:
            printf("Enter the name of the file that you would like to delete:\n");
            scanf("%s", name);
            if (searchFile(root, name) != NULL)
            {
                head = insertToList(head, name);
            }
            root = deleteFile(root, name);
            break;
        case 3:
            printf("Displaying all the files in the current directory:\n");
            inOrderTraversal(root);
            break;
        case 4:
            printf("Enter the name of the file you wish to search for:\n");
            scanf("%s", name);
            searchFile(root, name);
            break;
        case 5:
            printf("Enter the name of the file you would like to restore:\n");
            scanf("%s", name);
            restore = restoreDeletedFile(head, name);
            if (restore != NULL)
            {
                root = insertFile(root, restore->fname);
            }
            break;
        case 6:
            printf("Enter the extension to display files with that extension:\n");
            scanf("%s", name);
            displayFilesWithExtension(root, name);
            break;
        case 0:
            exit(0);
        default:
            break;
        }
    }
    printf("Printing all the files in the directory:\n");
    return 0;
}
```

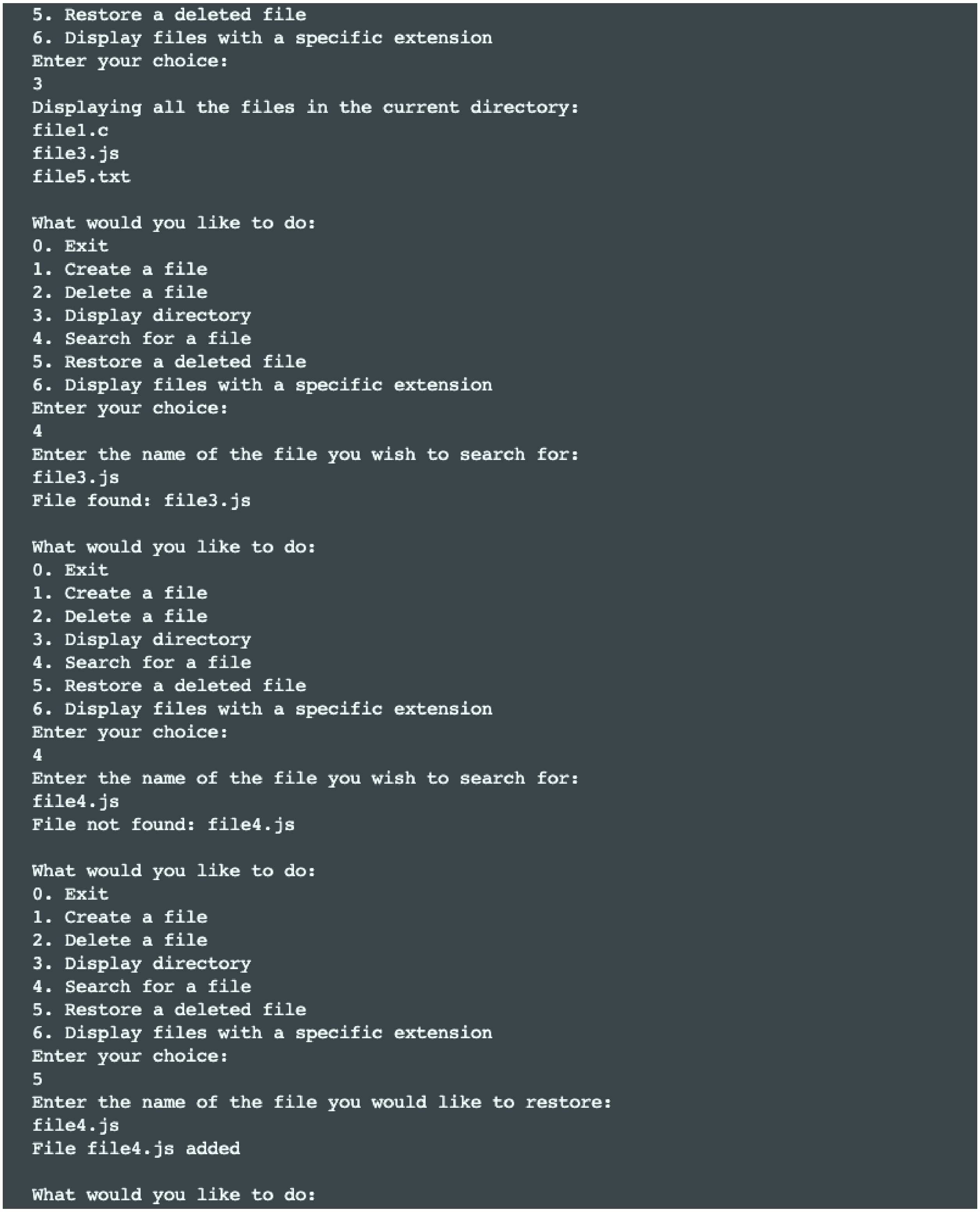
Output



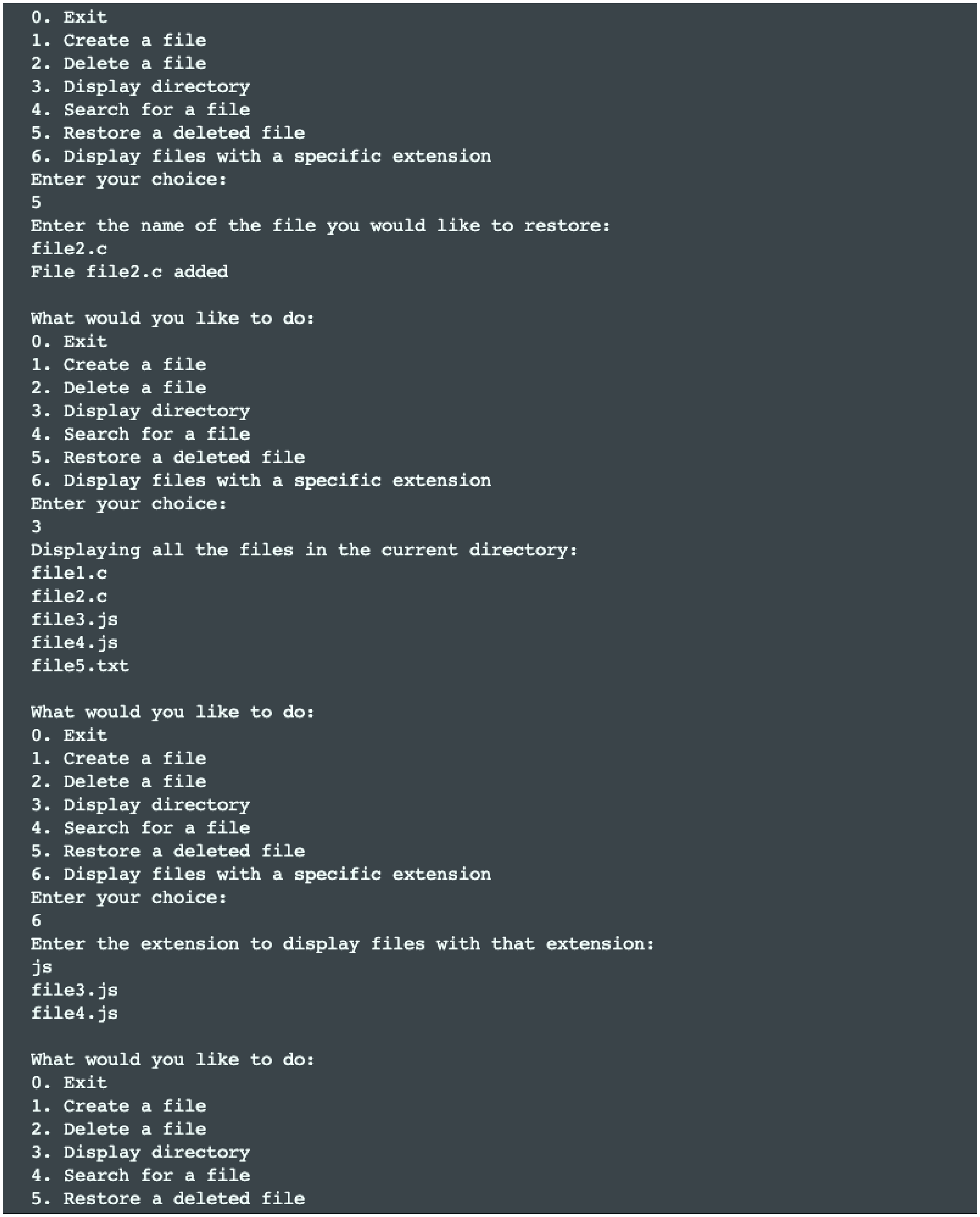
```
What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
Enter the name of the file that you would like to create:
file1.c
File file1.c added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
Enter the name of the file that you would like to create:
file2.c
File file2.c added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
Enter the name of the file that you would like to create:
file3.js
File file3.js added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
```



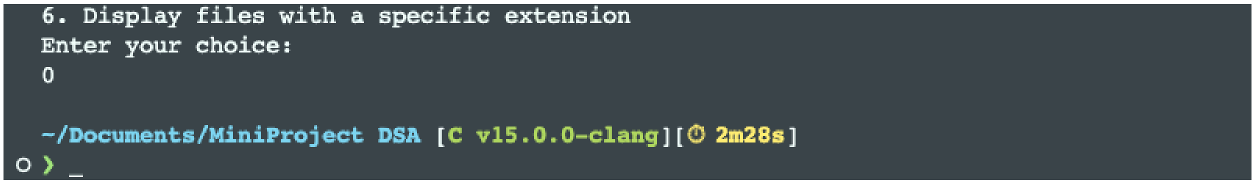
```
Enter the name of the file that you would like to create:
file4.js
File file4.js added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
Enter the name of the file that you would like to create:
file5.txt
File file5.txt added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
2
Enter the name of the file that you would like to delete:
file4.js
File file4.js deleted

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
2
Enter the name of the file that you would like to delete:
file2.c
File file2.c deleted

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
```



```
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
3
Displaying all the files in the current directory:
file1.c
file3.js
file5.txt

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
4
Enter the name of the file you wish to search for:
file3.js
File found: file3.js

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
4
Enter the name of the file you wish to search for:
file4.js
File not found: file4.js

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
5
Enter the name of the file you would like to restore:
file4.js
File file4.js added

What would you like to do:
```

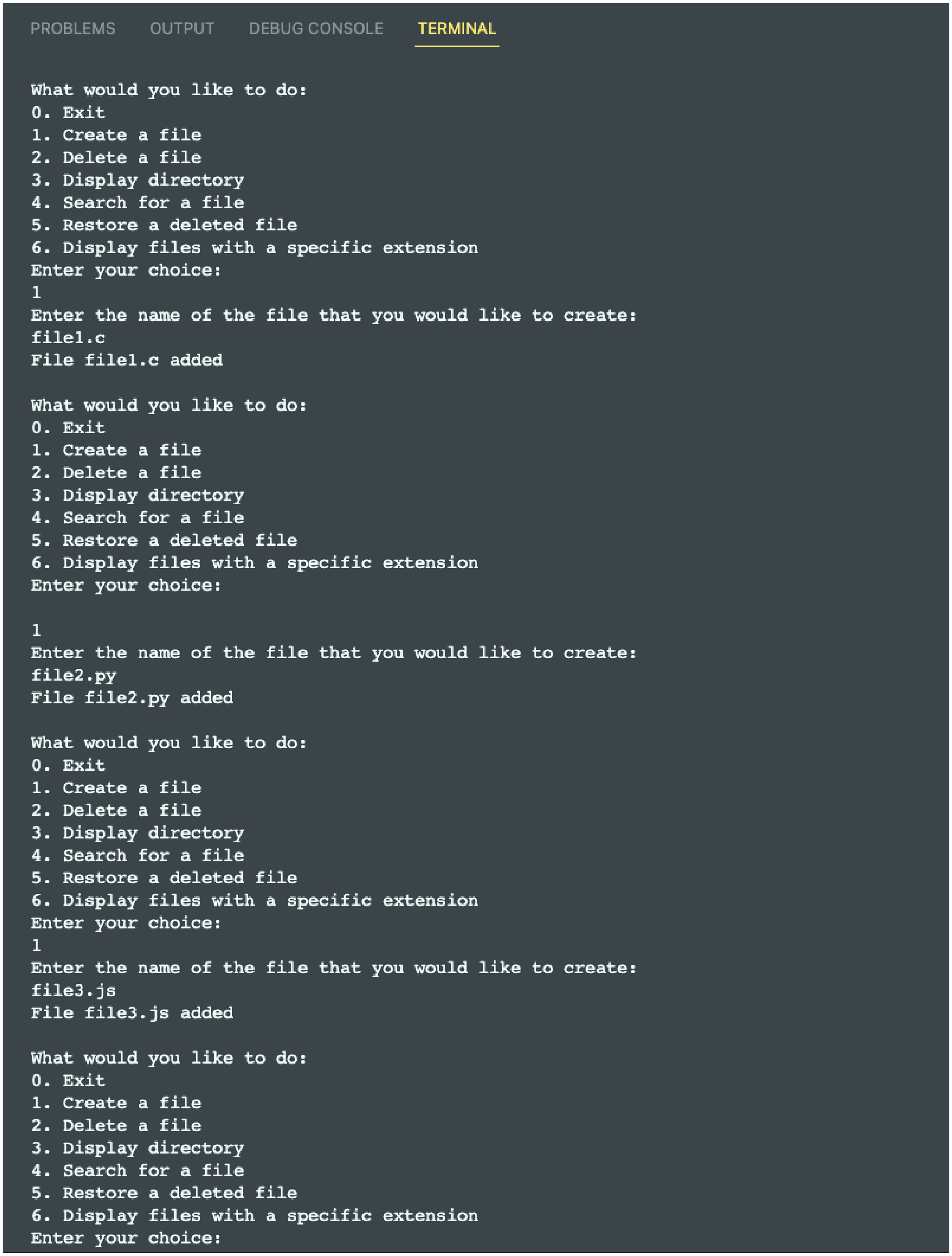


```
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
5
Enter the name of the file you would like to restore:
file2.c
File file2.c added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
3
Displaying all the files in the current directory:
file1.c
file2.c
file3.js
file4.js
file5.txt

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
6
Enter the extension to display files with that extension:
js
file3.js
file4.js

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
```

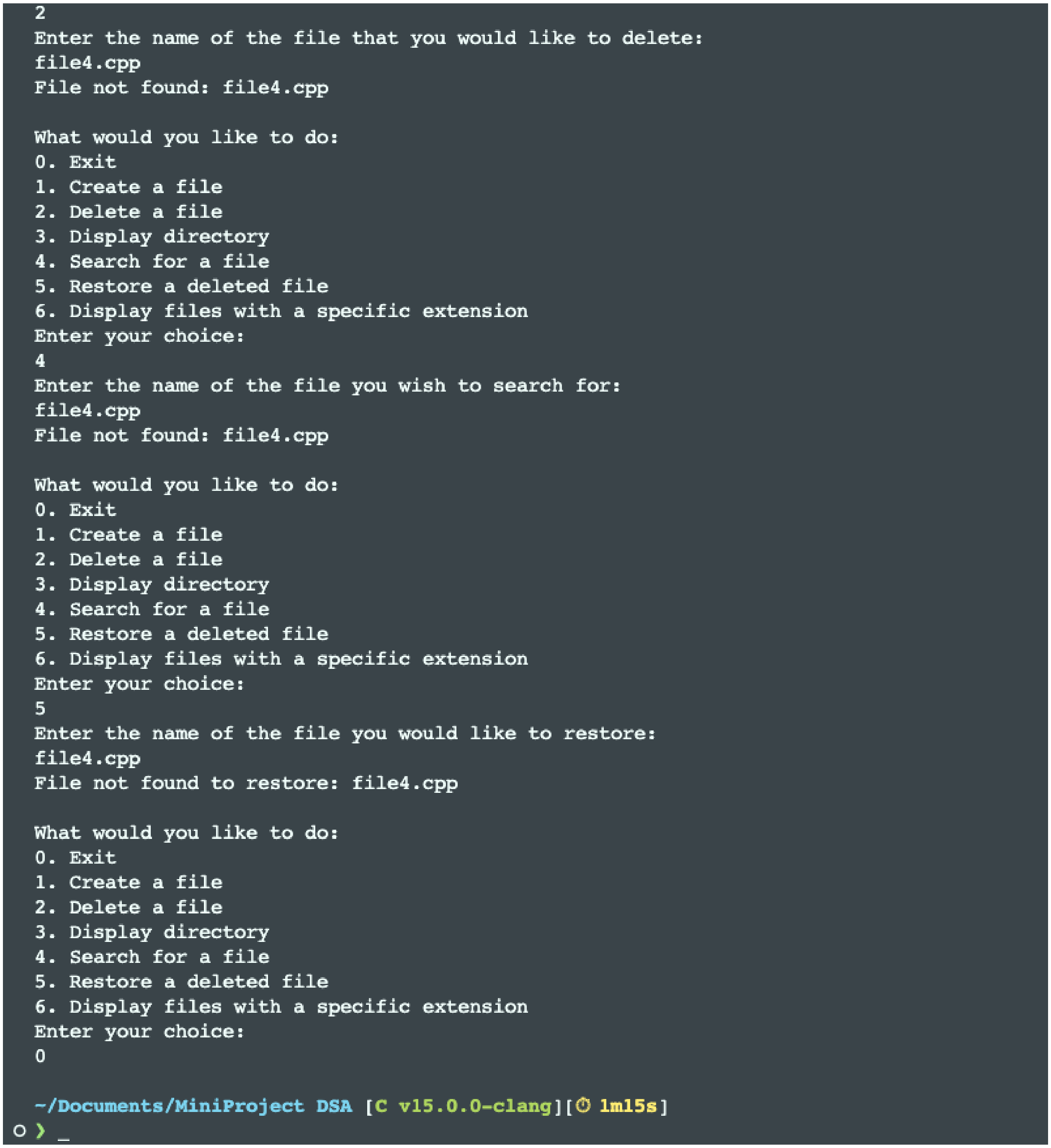


```
  6. Display files with a specific extension
  Enter your choice:
  0

  ~/Documents/MiniProject DSA [C v15.0.0-clang][⏱ 2m28s]
○ ❯ _
```

Output with Error Handling



```
What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
Enter the name of the file that you would like to create:
file1.c
File file1.c added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:

1
Enter the name of the file that you would like to create:
file2.py
File file2.py added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
1
Enter the name of the file that you would like to create:
file3.js
File file3.js added

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
```



```
2
Enter the name of the file that you would like to delete:
file4.cpp
File not found: file4.cpp

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
4
Enter the name of the file you wish to search for:
file4.cpp
File not found: file4.cpp

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
5
Enter the name of the file you would like to restore:
file4.cpp
File not found to restore: file4.cpp

What would you like to do:
0. Exit
1. Create a file
2. Delete a file
3. Display directory
4. Search for a file
5. Restore a deleted file
6. Display files with a specific extension
Enter your choice:
0

~/Documents/MiniProject DSA [C v15.0.0-clang][⏱ 1m15s]
○ ❯ _
```