EXP 14: Construct a C program to organize the file using single level directory.

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

#define MAX 100

struct Directory {
   char filename[100];
};

int main() {
   struct Directory dir[MAX];
   int count = 0;
   int choice;
   char name[100];

   while (1) {
      printf("\nSingle Level Directory\n");
      printf("1. Create File\n");
      printf("2. Delete File\n");
      printf("3. Search File\n");
      printf("4. Display Files\n");
      printf("5. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);
      getchar(); // To consume newline character

      switch (choice) {
```

```
case 1:
    if (count >= MAX) {
        printf("Directory full!\n");
        break;
    }
    printf("Enter file name to create: ");
    fgets(name, sizeof(name), stdin);
    name[strcspn(name, "\n")] = '\0';  // Remove newline

    // Check for duplicate
    int exists = 0;
    for (int i = 0; i < count; i++) {
        if (strcmp(dir[i].filename, name) == 0) {
            exists = 1;
            break;
        }
    }

    if (exists) {
        printf("File already exists.\n");
    } else {
        strcpy(dir[count].filename, name);
        count++;
        printf("File created successfully.\n");
    }
    break;

case 2:
    printf("Enter file name to delete: ");
    fgets(name, sizeof(name), stdin);
    name[strcspn(name, "\n")] = '\0';
```

```
    int found = 0;
    for (int i = 0; i < count; i++) {
        if (strcmp(dir[i].filename, name) == 0) {
            for (int j = i; j < count - 1; j++) {
                dir[j] = dir[j + 1];
            }
            count--;
            found = 1;
            printf("File deleted successfully.\n");
            break;
        }
    }

    if (!found)
        printf("File not found.\n");
    break;

case 3:
    printf("Enter file name to search: ");
    fgets(name, sizeof(name), stdin);
    name[strcspn(name, "\n")] = '\0';

    int is_found = 0;
    for (int i = 0; i < count; i++) {
        if (strcmp(dir[i].filename, name) == 0) {
            is_found = 1;
            printf("File found at position %d\n", i + 1);
            break;
        }
    }
```

```
            if (!is_found)
                printf("File not found.\n");
            break;

        case 4:
            if (count == 0)
                printf("Directory is empty.\n");
            else {
                printf("Files in directory:\n");
                for (int i = 0; i < count; i++) {
                    printf("%d. %s\n", i + 1, dir[i].filename);
                }
            }
            break;

        case 5:
            printf("Exiting...\n");
            exit(0);

        default:
            printf("Invalid choice!\n");
      }
   }

   return 0;
}
```

Sample Output

```
Single Level Directory
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice: 1
Enter file name to create: source
File created successfully.

Single Level Directory
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice: 1
Enter file name to create: source1
File created successfully.
```

```
Single Level Directory
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice: 2
Enter file name to delete: source1
File deleted successfully.

Single Level Directory
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice: 3
Enter file name to search: source
File found at position 1
```

```
Single Level Directory
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice: 4
Files in directory:
1. source

Single Level Directory
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice: 5
Exiting...
```