NAME: G. GANESH

REG NO: 192373008

EXERICSE-14

Construct a C program to organise the file using a single level directory.

Aim:

To construct a C program that simulates a **Single Level Directory** structure to manage files in an organized manner.

Algorithm:

1. Initialize Directory:
    - Create an array to store file names and a counter to track the number of files.
2. Menu Options:
    - Provide options to the user:
        1. Create a file.
        2. Delete a file.
        3. Search for a file.
        4. Display all files.
        5. Exit.
3. File Operations:
    - Create:
      Add the file name to the directory if it doesn't already exist.
    - Delete:
      Remove the file name from the directory if it exists.
    - Search:
      Check if the file exists in the directory.
    - Display:
      Show all files in the directory.
4. Exit the Program:
    - End the program when the user selects the exit option.

Code:

```
#include <stdio.h>

#include <string.h>
```

```
#define MAX_FILES 100
#define MAX_NAME_LENGTH 50
typedef struct {
    char name[MAX_NAME_LENGTH];
} File;
void create_file(File directory[], int *file_count) {
    if (*file_count >= MAX_FILES) {
        printf("Directory is full, cannot create more files.\n");
        return;
    }
    char file_name[MAX_NAME_LENGTH];
    printf("Enter file name to create: ");
    scanf("%s", file_name);
    for (int i = 0; i < *file_count; i++) {
        if (strcmp(directory[i].name, file_name) == 0) {
            printf("File '%s' already exists.\n", file_name);
            return;
        }
    }
    strcpy(directory[*file_count].name, file_name);
    (*file_count)++;
    printf("File '%s' created successfully.\n", file_name);
}
void delete_file(File directory[], int *file_count) {
    char file_name[MAX_NAME_LENGTH];
    printf("Enter file name to delete: ");
    scanf("%s", file_name);
    for (int i = 0; i < *file_count; i++) {
        if (strcmp(directory[i].name, file_name) == 0) {
            for (int j = i; j < *file_count - 1; j++) {
```

```
            directory[j] = directory[j + 1];
        }
        (*file_count)--;
        printf("File '%s' deleted successfully.\n", file_name);
        return;
    }
  }
  printf("File '%s' not found.\n", file_name);
}
void search_file(File directory[], int file_count) {
  char file_name[MAX_NAME_LENGTH];
  printf("Enter file name to search: ");
  scanf("%s", file_name);
  for (int i = 0; i < file_count; i++) {
    if (strcmp(directory[i].name, file_name) == 0) {
      printf("File '%s' found in the directory.\n", file_name);
      return;
    }
  }
  printf("File '%s' not found.\n", file_name);
}
void display_files(File directory[], int file_count) {
  if (file_count == 0) {
    printf("No files in the directory.\n");
    return;
  }
  printf("Files in the directory:\n");
  for (int i = 0; i < file_count; i++) {
    printf("%d. %s\n", i + 1, directory[i].name);
  }
```

```
}
int main() {
    File directory[MAX_FILES];
    int file_count = 0, choice;
    do {
        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);
        switch (choice) {
            case 1:
                create_file(directory, &file_count);
                break;
            case 2:
                delete_file(directory, &file_count);
                break;
            case 3:
                search_file(directory, file_count);
                break;
            case 4:
                display_files(directory, file_count);
                break;
            case 5:
                printf("Exiting the program.\n");
                break;
            default:
```

```
            printf("Invalid choice! Please try again.\n");
        }
    } while (choice != 5);
    return 0;
}
```

Result:

The program successfully simulates a single-level directory structure, allowing file creation, deletion, searching, and displaying the directory contents.

Output: