C LANGUAGE PROJECT

GROUP A

CODE OF DISASTER MANAGEMENT SYSTEM

**INPUT**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Structure to represent a disaster

struct Disaster {

    int id;

    char type[50];

    char location[100];

    char date[20];

};

// Function to add a new disaster

void addDisaster(struct Disaster \*disasters, int \*count) {

    printf("Enter disaster type: ");

    scanf("%s", disasters[\*count].type);

    printf("Enter location: ");

    scanf("%s", disasters[\*count].location);

    printf("Enter date: ");

    scanf("%s", disasters[\*count].date);

    disasters[\*count].id = \*count + 1; // Assigning a unique ID

    (\*count)++;

    printf("Disaster added successfully!\n");

}

// Function to display all disasters

void displayDisasters(const struct Disaster \*disasters, int count) {

    printf("ID\tType\tLocation\tDate\n");

    for (int i = 0; i < count; ++i) {

        printf("%d\t%s\t%s\t%s\n", disasters[i].id, disasters[i].type, disasters[i].location, disasters[i].date);

    }

}

int main() {

    struct Disaster disasters[100]; // Assuming a maximum of 100 disasters

    int count = 0; // To keep track of the number of disasters

    while (1) {

        // Display menu options

        printf("\nDisaster Management System\n");

        printf("1. Add Disaster\n");

        printf("2. Display Disasters\n");

        printf("3. Exit\n");

        printf("Enter your choice: ");

        int choice;

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                addDisaster(disasters, &count);

                break;

            case 2:

                displayDisasters(disasters, count);

                break;

            case 3:

                printf("Exiting the program.\n");

                exit(0);

            default:

                printf("Invalid choice. Please try again.\n");

        }

    }

    return 0;

}

**OUTPUT**

