

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

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

```
#define MAX_TASKS 10
```

```
#define MAX_TASK_NAME 100
```

```
typedef struct {
```

```
    char taskName[MAX_TASK_NAME];
```

```
    int isCompleted;
```

```
} Task;
```

```
Task tasks[MAX_TASKS];
```

```
int taskCount = 0;
```

```
void addTask() {
```

```
    if (taskCount < MAX_TASKS) {
```

```
        printf("Enter task name: ");
```

```
        fgets(tasks[taskCount].taskName, sizeof(tasks[taskCount].taskName), stdin);
```

```
        tasks[taskCount].taskName[strcspn(tasks[taskCount].taskName, "\n")] = 0;
```

```
        tasks[taskCount].isCompleted = 0;
```

```
        taskCount++;
```

```
        printf("Task added successfully!\n");
```

```
    } else {
```

```
        printf("Task list is full!\n");
```

```
    }
```

```
}
```

```
void viewTasks() {  
    printf("Task List:\n");  
    for (int i = 0; i < taskCount; i++) {  
        printf("%d. %s", i + 1, tasks[i].taskName);  
        if (tasks[i].isCompleted) {  
            printf(" (Completed)\n");  
        } else {  
            printf(" (Not Completed)\n");  
        }  
    }  
}
```

```
void markTaskAsCompleted() {  
    int taskNumber;  
    printf("Enter task number to mark as completed: ");  
    scanf("%d", &taskNumber);  
    if (taskNumber > 0 && taskNumber <= taskCount) {  
        tasks[taskNumber - 1].isCompleted = 1;  
        printf("Task marked as completed!\n");  
    } else {  
        printf("Invalid task number!\n");  
    }  
}
```

```

void editTask() {

    int taskNumber;

    printf("Enter task number to edit: ");

    scanf("%d", &taskNumber);

    if (taskNumber > 0 && taskNumber <= taskCount) {

        printf("Enter new task name: ");

        fgets(tasks[taskNumber - 1].taskName, sizeof(tasks[taskNumber - 1].taskName),
std in);

        tasks[taskNumber - 1].taskName[strcspn(tasks[taskNumber - 1].taskName, "\n")] = 0;

        printf("Task updated successfully!\n");

    } else {

        printf("Invalid task number!\n");

    }

}

```

```

void deleteTask() {

    int taskNumber;

    printf("Enter task number to delete: ");

    scanf("%d", &taskNumber);

    if (taskNumber > 0 && taskNumber <= taskCount) {

        for (int i = taskNumber - 1; i < taskCount - 1; i++) {

            strcpy(tasks[i].taskName, tasks[i + 1].taskName);

            tasks[i].isCompleted = tasks[i + 1].isCompleted;

        }

        taskCount--;

        printf("Task deleted successfully!\n");

    }

}

```

```
    } else {  
        printf("Invalid task number!\n");  
    }  
}
```

```
int main() {  
    int choice;  
    while (1) {  
        printf("To-Do List Menu:\n");  
        printf("1. Add Task\n");  
        printf("2. View Tasks\n");  
        printf("3. Mark Task as Completed\n");  
        printf("4. Edit Task\n");  
        printf("5. Delete Task\n");  
        printf("6. Exit\n");  
        printf("Enter your choice: ");  
        scanf("%d", &choice);  
        switch (choice) {  
            case 1:  
                addTask();  
                break;  
            case 2:  
                viewTasks();  
                break;  
            case 3:  
                markTaskAsCompleted();
```

```
        break;
    case 4:
        editTask();
        break;
    case 5:
        deleteTask();
        break;
    case 6:
        printf("Exiting...\n");
        return 0;
    default:
        printf("Invalid choice. Please try again.\n");
    }
}
return 0;
}
```