

```

// TO DO LIST 2.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <string>

struct Task {
    std::string description;
    bool completed;

    Task(const std::string& desc) : description(desc), completed(false) {}
};

class ToDoList {
private:
    std::vector<Task> tasks;

public:
    void AddTask(const std::string& desc) {
        tasks.emplace_back(desc);
        std::cout << "Task added: " << desc << std::endl;
    }

    void ViewTasks() {
        if (tasks.empty()) {
            std::cout << "No tasks in the list." << std::endl;
            return;
        }

        std::cout << "Tasks:" << std::endl;
        for (size_t i = 0; i < tasks.size(); ++i) {
            std::cout << "[" << (tasks[i].completed ? "X" : " ") << "]" << i + 1 << ". "
<< tasks[i].description << std::endl;
        }
    }

    void MarkTaskCompleted(int index) {
        if (index >= 1 && index <= tasks.size()) {
            tasks[index - 1].completed = true;
            std::cout << "Marked task " << index << " as completed." << std::endl;
        } else {
            std::cout << "Invalid task index." << std::endl;
        }
    }

    void RemoveTask(int index) {
        if (index >= 1 && index <= tasks.size()) {
            tasks.erase(tasks.begin() + index - 1);
            std::cout << "Removed task " << index << "." << std::endl;
        } else {
            std::cout << "Invalid task index." << std::endl;
        }
    }
};

int main() {
    ToDoList toDoList;

```

```

int choice;

while (true) {
    std::cout << "\nTo-Do List Manager" << std::endl;
    std::cout << "1. Add Task" << std::endl;
    std::cout << "2. View Tasks" << std::endl;
    std::cout << "3. Mark Task as Completed" << std::endl;
    std::cout << "4. Remove Task" << std::endl;
    std::cout << "5. Quit" << std::endl;
    std::cout << "Enter your choice: ";
    std::cin >> choice;

    switch (choice) {
        case 1: {
            std::string taskDescription;
            std::cout << "Enter task description: ";
            std::cin.ignore();
            std::getline(std::cin, taskDescription);
            toDoList.AddTask(taskDescription);
            break;
        }
        case 2: {
            toDoList.ViewTasks();
            break;
        }
        case 3: {
            int taskIndex;
            std::cout << "Enter the task index to mark as completed: ";
            std::cin >> taskIndex;
            toDoList.MarkTaskCompleted(taskIndex);
            break;
        }
        case 4: {
            int taskIndex;
            std::cout << "Enter the task index to remove: ";
            std::cin >> taskIndex;
            toDoList.RemoveTask(taskIndex);
            break;
        }
        case 5: {
            std::cout << "Goodbye!" << std::endl;
            return 0;
        }
        default: {
            std::cout << "Invalid choice. Please try again." << std::endl;
        }
    }
}

system("pause");
return 0;
}

```