#include <iostream>

#include <cstring>

using namespace std;

struct Task {

char description[100];

bool completed;

};

void addTask(Task tasks[], int& taskCount, const char\* description) {

if (taskCount < 10) {

Task newTask;

strcpy(newTask.description, description);

newTask.completed = false;

tasks[taskCount++] = newTask;

cout << "Task added successfully.\n";

} else {

cout << "Task list is full. Cannot add more tasks.\n";

}

}

void viewTasks(const Task tasks[], int taskCount) {

if (taskCount > 0) {

cout << "Tasks:\n";

for (int i = 0; i < taskCount; ++i) {

cout << i + 1 << ". " << tasks[i].description << " - "

<< (tasks[i].completed ? "Completed" : "Pending") << "\n";

}

} else {

cout << "No tasks available.\n";

}

}

// Function to mark a task as completed

void markAsCompleted(Task tasks[], int taskCount, int taskIndex) {

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

tasks[taskIndex].completed = true;

cout << "Task marked as completed.\n";

} else {

cout << "Invalid task index.\n";

}

}

// Function to remove a task

void removeTask(Task tasks[], int& taskCount, int taskIndex) {

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

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

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

}

--taskCount;

cout << "Task removed successfully.\n";

} else {

cout << "Invalid task index.\n";

}

}

int main() {

Task tasks[10];

int taskCount = 0;

char choice;

do {

cout << "\n===== To-Do List Manager =====\n";

cout << "1. Add Task\n";

cout << "2. View Tasks\n";

cout << "3. Mark Task as Completed\n";

cout << "4. Remove Task\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case '1': {

char description[100];

cout << "Enter task description: ";

cin.ignore(); // Clear the input buffer

cin.getline(description, sizeof(description));

addTask(tasks, taskCount, description);

break;

}

case '2':

viewTasks(tasks, taskCount);

break;

case '3': {

int taskIndex;

cout << "Enter task index to mark as completed: ";

cin >> taskIndex;

markAsCompleted(tasks, taskCount, taskIndex - 1);

break;

}

case '4': {

int taskIndex;

cout << "Enter task index to remove: ";

cin >> taskIndex;

removeTask(tasks, taskCount, taskIndex - 1);

break;

}

case '5':

cout << "Exiting the program. Goodbye!\n";

break;

default:

cout << "Invalid choice. Please try again.\n";

}

} while (choice != '5');

return 0;

}