#include <iostream>

#include <map>

#include <vector>

using namespace std;

// Task structure to store task details

struct Task {

string description;

};

// Calendar class to manage tasks

class Calendar {

private:

map<string, vector<Task> > tasks; // Map to store tasks for each date

public:

// Function to add a task to a specific date

void addTask(string date, string description) {

tasks[date].push\_back({description});

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

}

// Function to view tasks for a specific date

void viewTasks(string date) {

if (tasks.find(date) != tasks.end()) {

cout << "Tasks for " << date << ":\n";

for (size\_t i = 0; i < tasks[date].size(); ++i) {

cout << "- " << tasks[date][i].description << "\n";

}

} else {

cout << "No tasks for " << date << "\n";

}

}

};

int main() {

Calendar calendar;

while (true) {

cout << "\n===== Simple Task Manager =====\n";

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

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

cout << "3. Exit\n";

cout << "Enter your choice: ";

int choice;

cin >> choice;

switch (choice) {

case 1: {

string date, description;

cout << "Enter date (YYYY-MM-DD): ";

cin >> date;

cout << "Enter task description: ";

cin.ignore(); // Consume the newline character

getline(cin, description);

calendar.addTask(date, description);

break;

}

case 2: {

string date;

cout << "Enter date (YYYY-MM-DD): ";

cin >> date;

calendar.viewTasks(date);

break;

}

case 3:

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

return 0;

default:

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

}

}

return 0;

}