Midterm Project:

Travel Recommendations for Affluent Adventurers

Project Overview:

Kentucky is home to some with seemingly unlimited means. Many of them enjoy traveling to exotic destinations, some of which can surprisingly be reached without leaving the continental US. Although you will probably not have the good fortune of experiencing these destinations, you have been asked to create a schedule for a reporter who will get to visit these places and write about them. However, there are significant time and, yes, budget constraints.

For this project, you will be given a list of cities where the destinations are located, along with how much accommodations cost at each destination, and the travel time between each city. You will need to write a C++ program that maximizes the number of destinations visited, given specified time and budget constraints. (For the purposes of this project, the only expense you need to worry about is the cost of accommodations.) Who knows, if the total cost is low enough, you might get to go along with the reporter!

The input file will be in CSV format. The first line will contain two numbers followed by the name of a city, all separated by commas: the total amount of time in days – this will be a whole number - the reporter has to visit destinations, the maximum budget available for accommodations, and what city the reporter will be starting from. Each of the remaining lines in the file will either be the name of a destination followed by the cost of accommodations in US dollars, or the names of two cities and the travel time in hours (including fractional hours, if any) between them, with commas separating the values on each line. There won't be any blank lines in the input file. Assume that if no travel time is given between a pair of cities, there is no road connecting those cities.

The reporter will begin the journey at the starting location specified on the first line in the input file at 11 AM on the first day, and he or she can travel up to 8 hours each day. The journey will end at 11 AM at the final destination, regardless of how far it is from the starting location. If the travel time from one destination to the next is over 8 hours, travel will need to be broken up in such a way that no more than 8 hours are traveled on any given day. Accommodations when travel is broken up in this fashion will cost $100 per night until the destination is reached.

Your program will need to output the recommended route along with:

- The travel time to each destination (from the previous destination)
- The cost of accommodations at each destination visited, including the cost of accommodations in case travel from one destination to the next had to be broken up into more than one day
- The total cost of accommodations for the entire journey
- The total number of days (rounded up, if necessary) the journey will take from beginning to end

If your team is feeling particularly ambitious, your program could try to find the route that minimizes total time and cost. However, if there is more than one route with the same number of destinations, your program will receive full credit if it only considers one of them.

Expectations:

- Your program needs to be readable – it needs to be possible for someone reading your program to follow along what your program is doing.
- You will be provided with a C++ program to use as your starting point. However, your program must be a demonstration of how to solve this problem using object-oriented programming.
- Each person in your team needs to be responsible for writing at least one class declaration and implementation. In addition, each person in your team needs to write a program that tests at least one class implementation that someone else in the team wrote.
- Each team will need to create two or three data files that demonstrate that their program works correctly. These data files need to conform to the specifications above, and they need to be of value. In other words, at a minimum, they need to be useful for determining if the program generally works correctly. Each data file needs to have associated expected output. Grading will take into account how completely the data files test program functionality.

Due Date:

- February 25 – final, production ready program and data files – late submissions will not be allowed!

Team Composition:

All but two teams will consist of three people chosen at random. The remaining teams will consist of four people, also chosen at random.

Class Presentations:

Each team will give a presentation on February 25th or 26th on how they approached the problem and how their code works, as well as demonstrate the program. Each team member must participate in the presentation and talk about the code he or she wrote, as well as how he or she tested the code his or her team member wrote. Although the presentation will need to address what was done, it should focus on how the code works and why it was written that way.