College Tour

Your team is to write a software that reads data and stores it in a data structure of your choice. Your program must have the ability to hold the names of at least 20 college campuses (the initial data contains 11 campuses) and their corresponding traditional souvenirs (up to seven per campus). Your program will be used to allow perspective college students to plan their college experience. Your team will need to provide the ability to modify information related to the traditional souvenirs (such as adding new souvenirs, changing a price of a souvenir, or deleting a souvenir).

Displaying the initial information:

1. Display the list of college campuses and their distances from Saddleback College.

2. Display all the traditional souvenirs for any given campus.

Planning a trip:

3. Provide the capability to visit the initial 11 college campuses starting at University of California, Irvine (UCI).
   a. Plan the trip starting at UCI then visit each of the other college campuses in the most efficient order (recursively choose the campus closest to the previous campus)
   b. Display the total distance traveled

4. Offer the option to plan the **shortest** trip starting at Arizona State University (ASU).
   a. Obtain the "number" of college campuses to visit
   b. Visit the "number" of college campuses specified (including at Arizona State University (ASU))
   c. Plan the trip starting at Arizona State University (ASU) then visit the closest college campuses to ASU, then visit the closest college campus to that college and so on (recursively choose the campus closest to the previous campus)
   d. Display the total distance traveled

e. Allow the college student to purchase multiple traditional souvenirs when visiting the college campuses

5. Offer the option to plan a custom trip
   a. Allow a college student to select the starting college campus they wish to visit
   b. Then allow a college student to select <u>all</u> other college campuses they wish to visit
   c. Plan the trip starting with the selected campus then visit each of the other college campuses in the most efficient order (recursively choose the campus closest to the previous campus).
   d. Display the total distance traveled
   e. Allow the college student to purchase multiple traditional souvenirs when visiting the college campuses

6. When taking <u>any</u> trip:
   a. A college student can purchase multiple traditional souvenirs
   b. Your team must keep track of the number of traditional souvenirs purchased at each campus
   c. Display the total amount spent at each college campus and a grand total for all campuses visited

7. Maintenance (administrator only - requires a password to gain access)
   a. Provide the capability to add new college campuses and their corresponding souvenirs by having your program read from an input file given to the class
   b. Provide the capability to change the prices of the traditional souvenirs
   c. Provide the capability to add new traditional souvenirs
   d. Provide the capability to delete traditional souvenirs

8. Provide the capability to visit the 13 college campuses starting at Saddleback.
   a. Display the total distance traveled

b. Allow the college student to purchase multiple traditional souvenirs when visiting the college campuses

Please let me know your partners by January 29th (three points will be deducted from your scores if you do not meet this deadline). All projects are due by March 23th. **No late projects will be accepted.** Your team must demonstrate your project to me before it will be graded. Each teammate must identify their accomplishments on the project. Not all team members will necessarily earn the same score.

1. Design a very readable, easy to use interface to demonstrate your program.
2. Contingency handling should include addressing invalid input.
3. Write at least 10 agile stories (including description, tasks, test scenarios, and story points) before any software is developed. The team must follow the Scrum process (the Scrum master **must** document all meetings and the product owner must document the backlog).
4. Submit the following UML class diagrams
   a. Three use cases
   b. One activity diagram
   c. Three state diagrams with your project.
5. Submit a test plan.
6. All changes must be persistent between executions.
7. Submit a discussion the **Big-Oh** of your project for at least **five** methods.
8. Identify all the data structures used
9. Each team must use a version control system (only team members should have access to their repository), graphical user interface tool, automated documentation tool, and an Agile management tool. (GITHUB, DOXYGEN, WAFFIO.IO, graphical user interface using software such as QT, etc.).

Schedule:
First checkpoint – February 24th
Second checkpoint – March 9th
Final checkpoint – March 23rd

The project will be graded using the following scale:

	Value
Checkpoint 1	4
Checkpoint 2	4
Meet requirements	65
Coding Style/Style Guide	2
Use of multiple C++ data structures (2 points per data structure)	6
User interface	2
Test Plan	3
Adherence to Scrum/Team Rules	4
UML	3
DOXYGEN	1
Big O (1 point for each method)	5
Contingency handling	1
Total	100
Continuous Integration (extra credit)	3
Total with extra credit	103

Final demonstration meeting:

1. Be prepared to demonstrate all project's requirements within the 20-minute timeframe.
2. All team members must be present.
3. Turn in Agile planning poker cards
4. Demonstrate DOXYGEN and Agile management tool
5. Each teammate must identify their accomplishments on the project and assess their teammates via e-mail.
6. Submit all your project artifacts
   a. Code, test plan, agile stories, scrum log, coding standards, team rules, UML diagrams (class, use cases, state diagrams, data structures used, Big Oh analysis