Your team is to write a program that reads data and stores it in a data structure of your choice. The data structure must have the ability to hold information at least 30 wineries (the initial data contains fewer than that) where each winery may have a maximum of 10 different wines. This data structure will be used to allow guests at The Canyon Villa in Paso Robles, CA to plan various day trips. You must be able to randomly access the various wineries. You will need to provide the ability to modify information about a winery (such as adding wines to the existing list), print a list of the wineries, plan a trip, and take a trip.

**Planning a trip:**

1. Output the list of wineries and their distance from The Canyon Villa
2. Offer the option to plan a custom trip or have the shortest trip planned
3. If shortest trip (most efficient really)
   1. Allow a user to select a beginning winery
   2. Obtain the number of wineries to visit
   3. Display the list of wineries visited
   4. Display the total distance traveled
4. If custom trip
   1. Allow a user to select all wineries they wish to visit
   2. Plan the trip starting with the selected winery closest to The Canyon Villa then visit each of the other wineries in the most efficient order (the least amount of driving mileage)
   3. Display the total distance traveled
5. Have an option to visit all the wineries
   1. Display the total distance traveled
6. When taking a trip:
   1. A user can shop for wines
   2. You must keep track of the number of bottles of wine purchased and the total spent at each winery (plus a grand total)
7. Maintenance (administrator only - requires a pass code to gain access)
   1. Add new wineries
   2. Add new wines to existing winery
   3. Change prices of wines

Sample main menu:

STRAIGHT AHEAD WINE TOURS OF THE CENTRAL COAST

1. View our list of wineries

2. Plan a day trip

3. Tour your wineries and purchase wines

4. Perform file maintenance (administrator only)

5. Quit

Please let me know who your partners will be by January 29th (two points will be deducted from your score if you do not meet this deadline). All projects are due by March 17th. 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.

Design a very readable, easy to use interface to demonstrate your program. Contingency handling should include addressing invalid input.

Submit an UML class diagram with your project. Submit a paragraph discussing the Big-Oh of your project. Identify all the data structures used. Each team must use a version control system (GIT, Jazzhub, GITHUB, etc.) and QT.

Schedule:

First checkpoint – February 19th – 5 points

Second checkpoint – March 5th – 5 points

Final checkpoint – March 17th – 15 points

The assignment will be graded using the following scale:

|  |  |
| --- | --- |
|  | Value |
| Meet requirements | 8 |
| Style | 3 |
| Use of C++ data structures | 4 |
| User interface | 3 |
| Contingency handling | 2 |
| Adherence to Scrum | 3 |
| Identification of Big Oh | 2 |
| Total | 25 |

**Final demonstration meeting**:

1. Submit all your project artifacts (code, agile stories, UML diagrams, traceability between agile stories and requirements, identify data structures used, identify accomplishments, etc.)
2. Be prepared to demonstrate all project’s requirements within the 30 minute timeframe.
3. All team members must be present.
4. Each individual team member must clearly identify their accomplishments.