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 for at least 20 fast food restaurants (the initial data contains fewer than that) where each fast food restaurant may have a maximum of 8 different menu items. This data structure will be used to allow foodies at Saddleback College to plan various day trips. You must be able to randomly access the various fast food restaurants. You will need to provide the ability to modify information about a fast food restaurant (such as adding menu items, changing a price of an item, or deleting an item). You should be able to print a list of the fast food restaurants, plan a trip, and take a trip.

**Planning a trip:**

1. Output the list of fast food restaurants and their distance from Saddleback College (the initial 10 restaurants)
2. Offer the option to plan the **shortest** trip starting at Saddleback College.
   1. Allow a user to select a beginning fast food restaurant
   2. Obtain the number of fast food restaurants to visit
   3. Visit the number of restaurants specified (including the beginning restaurant)
   4. Plan the trip starting with the selected fast food restaurant then visit each of the other fast food restaurants in the most efficient order (recursively choose the restaurant closest to the previous restaurant).
   5. Display the total distance traveled
   6. Allow the user to purchase menu items when visiting the fast food restaurants
3. Offer the option to plan a custom trip
   1. Allow a user to select all fast food restaurants they wish to visit
   2. Plan the trip starting with the selected fast food restaurant closest to Saddleback College then visit each of the other fast food restaurants in the most efficient order (recursively choose the restaurant closest to the previous restaurant). The custom trip should start at Saddleback College.
   3. Display the total distance traveled
   4. Allow the user to purchase menu items when visiting the fast food restaurants
4. Have an option to visit the initial 10 fast food restaurants starting at Saddleback College. (Also have the ability to visit additional restaurants after they are added)
   1. Plan the trip starting with the selected fast food restaurant closest to Saddleback College then visit each of the other fast food restaurants in the most efficient order (recursively choose the restaurant closest to the previous restaurant).
   2. Display the total distance traveled
   3. Allow the user to purchase menu items when visiting the fast food restaurants
5. When taking any trip:
   1. A user can shop for menu items
   2. You must keep track of the number of menu items purchased and the total spent at each fast food restaurant (plus a grand total)
   3. Output the total revenue for each fast food restaurant including a grand total for all the fast food restaurants
6. Maintenance (administrator only - requires a password to gain access)
   1. Add new fast food restaurants and the corresponding menu items from the input file given to the class
   2. Add the new restaurants from user input
   3. Change prices of menu items
   4. Adding new menu items
   5. Delete a menu item

Please let me know your partners by September 7th (two points will be deducted from your score if you do not meet this deadline). All projects are due by October 24th. **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 a UML class diagram, at least three use cases, and at least three state diagrams with your project.
5. Submit a test plan.
6. All changes must be persistent between executions.
7. Submit a paragraph discussing the **Big-Oh** of your project. Identify all the data structures used.
8. Each team must use a version control system, 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 – September 26st – 5 points

Second checkpoint – October 10th – 5 points

Final checkpoint – October 24th or – October 26th - 30 points

The project will be graded using the following scale:

****

**Final demonstration meeting**:

1. Submit all your project artifacts (code, test plan, 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 20 minute timeframe.
3. All team members must be present.
4. Each individual team member must clearly identify their accomplishments.