

Title: Restaurant ordering system [total 19 marks]

****DO NOT USE VECTOR**

Problem: create a program that will read restaurant's menu and display it, the user can use serial number to order and will also require how many customer there is and count how much each customer spend in this meal. The program will also store all the order taken in 24 hours and give a report when exit the program.

Mark Description:

1. Menu(including the choice such as show the menu and exit) [marks 0.5]

Menu example:

- 1.display the menu
- 2.Order(Enter customer number & serial number to order)
- 3.Bill(display bill and split the bill)
- 4.exit

2. Open and read the file(menu) [marks 1]

Open and read the csv file(menu)

3. Display the menu [marks 2]

4. Requires the user to enter customer number and serial number to order [marks 3.5]

When the user chose number 2 in the menu The program should ask the customer number and take the serial number that the user enter to calculate and add up the total cost

5. Return the bill and how much each person spend [marks 3.5]

The program should display the bill when the user chose 3 including what the customer ordered and the total bill and how much each person spend. After that the program should jump to the store step

6. Store every order within 24 hour [marks 4]

The program should store all of the order's income and the total customer number in 24 hours (it's the best if the program could also save all the item that was ordered in those order) after this step the program should open a new order when the user chose 2 again.

7. When hit exit return total income and how much each person spend per that day marks[4.5]

When the user exit the program the program will return out the total income within 24 hour and how much each person spend in that 24 hour. Then the program will end.

Must implement (but not limited to)

1. File handling (read and write)
2. struct
3. class
4. public and private access modifiers
5. Pointers
6. Functions
7. Passing pointers and “address of” to function(s)
8. template

Must have algorithms (but not limited to)

1. Double or Circular LinkedList
2. Binary Search
3. Priority Queue
4. Research and implement something not covered in class i. Binary Search Tree
ii. One of the sorting algorithms (not bubble, insertion, selection, merge, quicksort)

Other requirements

1. .h and .cpp files
2. Working Makefile
3. Proper use of const
4. Outputs should be properly formatted
5. Proper naming convention
6. Proper indentation
7. Restrict a file to one purpose. • Avoid dumping everything in 1 one file
8. Restrict the class to one purpose