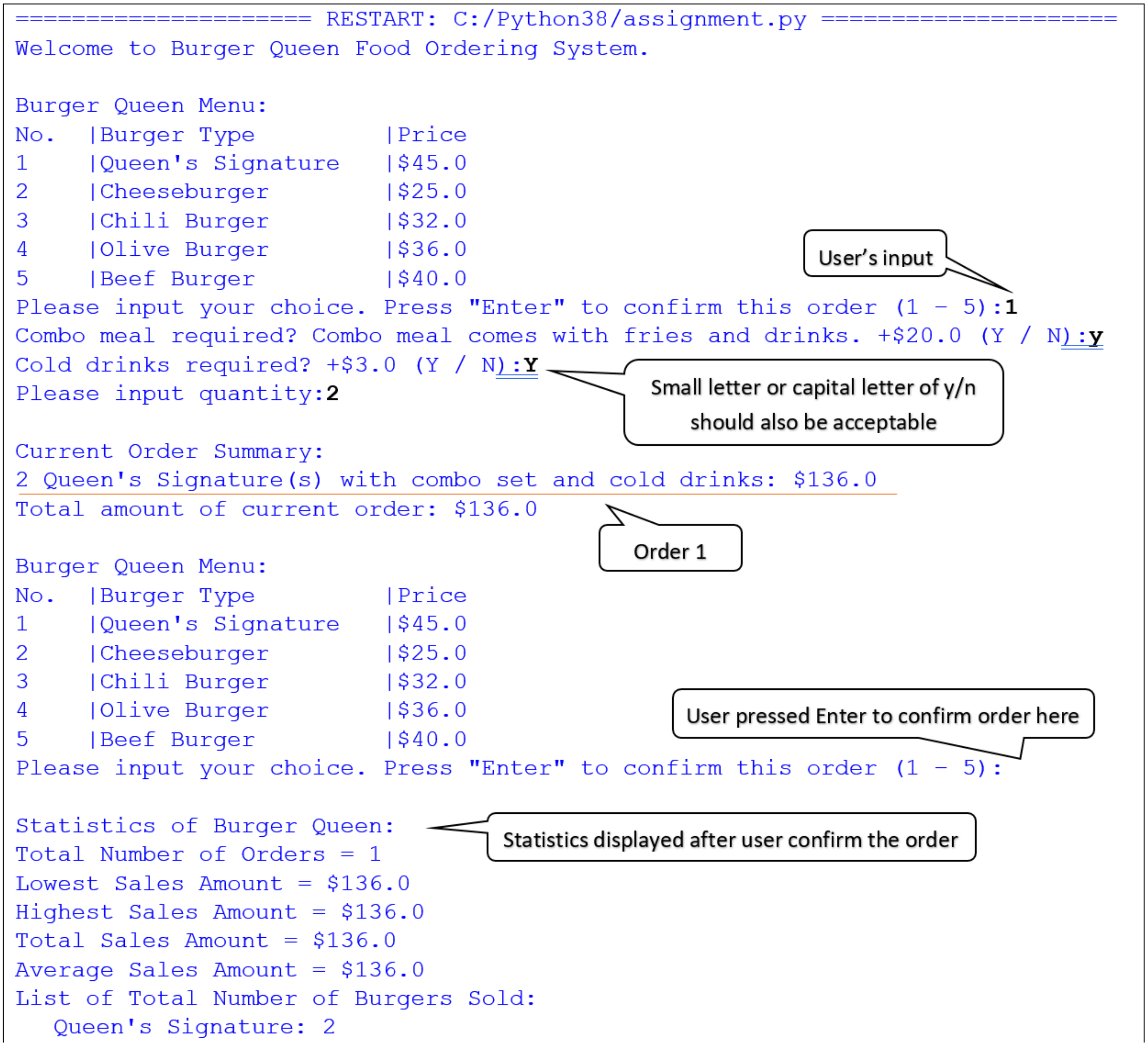
Assignment

Write a program to compute the sales for Burger Queen. The requirements are listed below:

- Burger Queen sells different types of burgers. In this season five types of burgers are offered: Queen's Signature, Cheeseburger, Chili Burger, Olive Burger and Beef Burger.
- A customer may pay extra money for a "combo meal" option.
- A customer may further choose and/or with “cold drink” when buying a burger with combo meal option
- The user is allowed to enter the quantity and choose the preferred options by answering Y or N,
- Display the current order summary including types of burgers, quantity, options requested and total amount when the user purchases all the burgers that s/he wants.
- Display sales statistics after user confirms his / her order.

A sample output is shown below:



```
===================== RESTART: C:/Python38/assignment.py =====================
Welcome to Burger Queen Food Ordering System.

Burger Queen Menu:
No.  |Burger Type         |Price
1    |Queen's Signature   |$45.0
2    |Cheeseburger        |$25.0
3    |Chili Burger        |$32.0
4    |Olive Burger        |$36.0
5    |Beef Burger         |$40.0
Please input your choice. Press "Enter" to confirm this order (1 – 5):1
Combo meal required? Combo meal comes with fries and drinks. +$20.0 (Y / N):y
Cold drinks required? +$3.0 (Y / N):Y
Please input quantity:2

Current Order Summary:
2 Queen's Signature(s) with combo set and cold drinks: $136.0
Total amount of current order: $136.0

Burger Queen Menu:
No.  |Burger Type         |Price
1    |Queen's Signature   |$45.0
2    |Cheeseburger        |$25.0
3    |Chili Burger        |$32.0
4    |Olive Burger        |$36.0
5    |Beef Burger         |$40.0
Please input your choice. Press "Enter" to confirm this order (1 – 5):

Statistics of Burger Queen:
Total Number of Orders = 1
Lowest Sales Amount = $136.0
Highest Sales Amount = $136.0
Total Sales Amount = $136.0
Average Sales Amount = $136.0
List of Total Number of Burgers Sold:
   Queen's Signature: 2
```

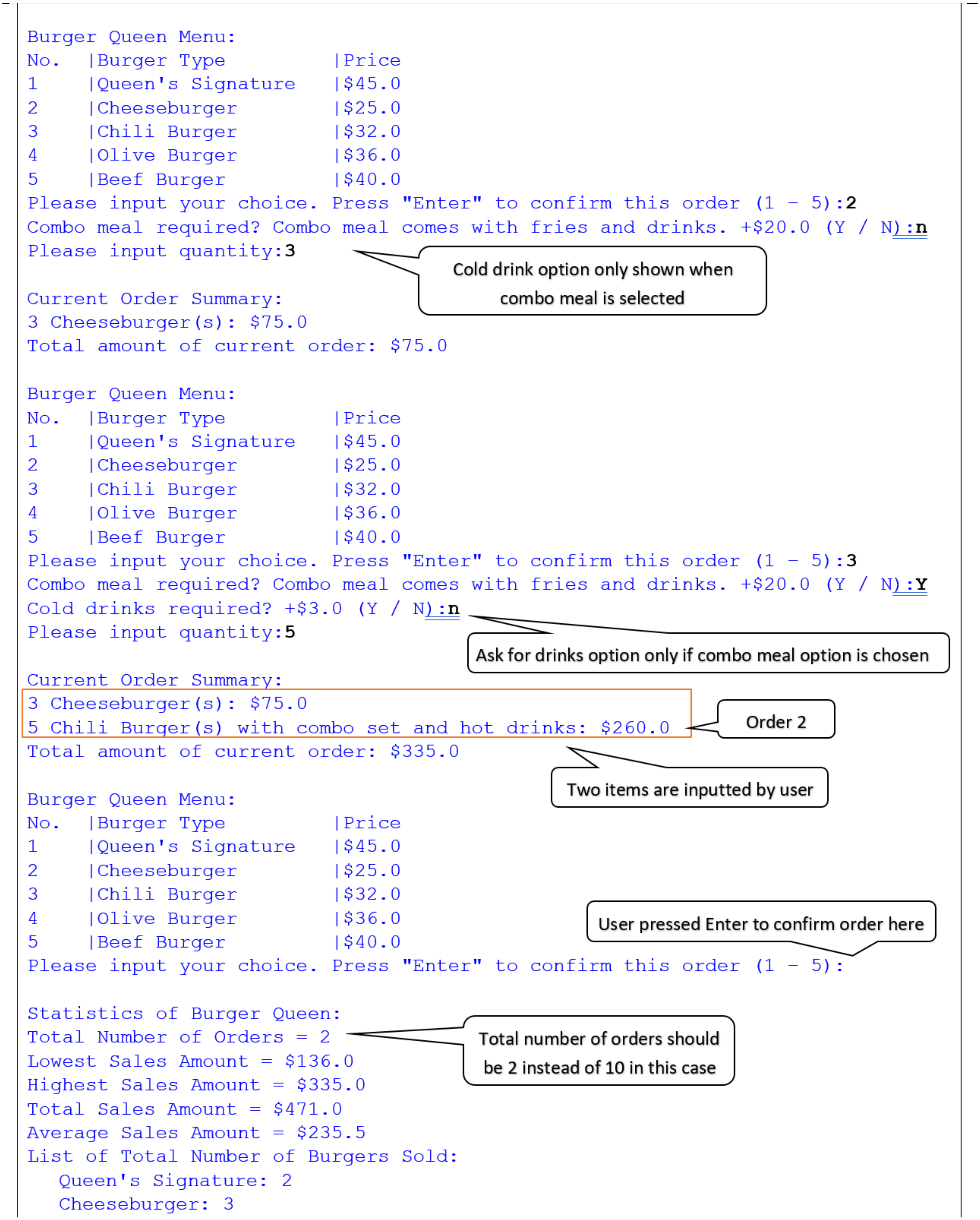
User’s input

Small letter or capital letter of y/n should also be acceptable

Order 1

User pressed Enter to confirm order here

Statistics displayed after user confirm the order



```
Burger Queen Menu:
No.  |Burger Type         |Price
1    |Queen's Signature   |$45.0
2    |Cheeseburger        |$25.0
3    |Chili Burger        |$32.0
4    |Olive Burger        |$36.0
5    |Beef Burger         |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):2
Combo meal required? Combo meal comes with fries and drinks. +$20.0 (Y / N):n
Please input quantity:3

Current Order Summary:
3 Cheeseburger(s): $75.0
Total amount of current order: $75.0

Burger Queen Menu:
No.  |Burger Type         |Price
1    |Queen's Signature   |$45.0
2    |Cheeseburger        |$25.0
3    |Chili Burger        |$32.0
4    |Olive Burger        |$36.0
5    |Beef Burger         |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):3
Combo meal required? Combo meal comes with fries and drinks. +$20.0 (Y / N):Y
Cold drinks required? +$3.0 (Y / N):n
Please input quantity:5

Current Order Summary:
3 Cheeseburger(s): $75.0
5 Chili Burger(s) with combo set and hot drinks: $260.0
Total amount of current order: $335.0

Burger Queen Menu:
No.  |Burger Type         |Price
1    |Queen's Signature   |$45.0
2    |Cheeseburger        |$25.0
3    |Chili Burger        |$32.0
4    |Olive Burger        |$36.0
5    |Beef Burger         |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):

Statistics of Burger Queen:
Total Number of Orders = 2
Lowest Sales Amount = $136.0
Highest Sales Amount = $335.0
Total Sales Amount = $471.0
Average Sales Amount = $235.5
List of Total Number of Burgers Sold:
   Queen's Signature: 2
   Cheeseburger: 3
```

Cold drink option only shown when combo meal is selected

Ask for drinks option only if combo meal option is chosen

Order 2

Two items are inputted by user

User pressed Enter to confirm order here

Total number of orders should be 2 instead of 10 in this case

```
   Chili Burger: 5

Burger Queen Menu:
No.   |Burger Type          |Price
1     |Queen's Signature    |$45.0
2     |Cheeseburger         |$25.0
3     |Chili Burger         |$32.0
4     |Olive Burger         |$36.0
5     |Beef Burger          |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):
Current Sales Order is empty

Burger Queen Menu:
No.   |Burger Type          |Price
1     |Queen's Signature    |$45.0
2     |Cheeseburger         |$25.0
3     |Chili Burger         |$32.0
4     |Olive Burger         |$36.0
5     |Beef Burger          |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):
```

User pressed Enter to confirm order here

User confirmed order without any items

sample output of exceptional case is shown below. A message "**File "burger-list.csv" cannot be opened**" should be splayed if the burger list cannot be read from the csv file.

```
Burger Queen Menu:
No.   |Burger Type          |Price
1     |Queen's Signature    |$45.0
2     |Cheeseburger         |$25.0
3     |Chili Burger         |$32.0
4     |Olive Burger         |$36.0
5     |Beef Burger          |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):0
Invalid input for choice
Please input your choice. Press "Enter" to confirm this order (1 - 5):2
Combo meal required? Combo meal comes with fries and drinks. +$20.0 (Y / N):a
Invalid input for choice
Combo meal required? Combo meal comes with fries and drinks. +$20.0 (Y / N):y
Cold drinks required? +$3.0 (Y / N):
Invalid input for choice
Cold drinks required? +$3.0 (Y / N):Y
Please input quantity:2.5
Invalid input for choice
Please input quantity:a
Invalid input for choice
Please input quantity:-1
Invalid input for choice
Please input quantity:0
Invalid input for choice
Please input quantity:2
```

Invalid function number

Invalid empty input

Invalid input. Only Y / N is accepted

Only integer value is accepted

```
Current Order Summary:
2 Cheeseburger(s) with combo set and cold drinks: $96.0
Total amount of current order: $96.0

Burger Queen Menu:
No.  |Burger Type        |Price
1    |Queen's Signature  |$45.0
2    |Cheeseburger       |$25.0
3    |Chili Burger       |$32.0
4    |Olive Burger       |$36.0
5    |Beef Burger        |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):
```

Follow the steps below to implement the program:

1. Declare two **constant variables** named **COMBO_MEAL_PRICE** and **COLD_DRINK_PRICE** to represent the price of each extra option (see the prices shown in the sample output above)

2. Complete the function **read_burger_from_file** to read in burgers information from file named “**burger_list.csv**” located in the same folder. The sample file “**burger_list.csv**” is given for you already and should follow the same format. **Zero marks** will be given if you assigned the burger data by static variable such as **list_dict_burger = [{…}]** as given.

3. Declare a **list of dictionary** named **list_dict_burgers** in main function to store all burgers information returned by the function **read_burger_from_file()**. You are NOT allowed to make the **list_dict_burgers** variable global. The following diagrams show the detailed structure of **list_dict_burgers**:

List of Dictionary **list_dict_burgers**

<table>
<tr><td>0</td><td>burger 1 record
<table><tr><td>Key</td><td>Data Type</td><td>Sample Value</td></tr><tr><td>Name</td><td>String</td><td>"Queen's Signature"</td></tr><tr><td>Price</td><td>Float</td><td>45.0</td></tr></table></td></tr>
<tr><td>1</td><td>burger 2 record
<table><tr><td>Key</td><td>Data Type</td><td>Sample Value</td></tr><tr><td>Name</td><td>String</td><td>"Cheeseburger"</td></tr><tr><td>Price</td><td>Float</td><td>25.0</td></tr></table></td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>n</td><td>burger n record
<table><tr><td>Key</td><td>Data Type</td><td>Sample Value</td></tr><tr><td>Name</td><td>String</td><td>"Beef Burger"</td></tr><tr><td>Price</td><td>Float</td><td>40.0</td></tr></table></td></tr>
</table>

Sample value of **list_dict_burgers**

```
[ {'Name': "Queen's Signature", 'Price': 45.0},
  {'Name': 'Cheeseburger', 'Price': 25.0},
  {'Name': 'Chili Burger', 'Price': 32.0},
  {'Name': 'Olive Burger', 'Price': 36.0},
  {'Name': 'Beef Burger', 'Price': 40.0} ]
```

4. Declare a **list of tuple** named **list_tuple_current_sales** in main function to store all burger orders which are not yet completed by customer. You are NOT allowed to make the **list_tuple_current_sales** variable global. The following diagrams show the detailed structure of **list_tuple_current_sales**:

List of Tuple **list_tuple_current_sales**

<table>
<tr><td>0</td><td>1st order
<table><tr><td>Index</td><td>Data Type</td><td>Sample Value</td></tr><tr><td>0</td><td>Integer</td><td>1</td></tr><tr><td>1</td><td>String</td><td>"Y"</td></tr><tr><td>2</td><td>String</td><td>"Y"</td></tr><tr><td>3</td><td>Integer</td><td>2</td></tr></table></td></tr>
<tr><td>1</td><td>2nd order
<table><tr><td>Index</td><td>Data Type</td><td>Sample Value</td></tr><tr><td>0</td><td>Integer</td><td>2</td></tr><tr><td>1</td><td>String</td><td>"Y"</td></tr><tr><td>2</td><td>String</td><td>"N"</td></tr><tr><td>3</td><td>Integer</td><td>3</td></tr></table></td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>n</td><td>nth order
<table><tr><td>Index</td><td>Data Type</td><td>Sample Value</td></tr><tr><td>0</td><td>Integer</td><td>4</td></tr><tr><td>1</td><td>String</td><td>"N"</td></tr><tr><td>2</td><td>String</td><td>"N"</td></tr><tr><td>3</td><td>Integer</td><td>2</td></tr></table></td></tr>
</table>

Sample value of **list_tuple_current_sales**

```
[(1, 'Y', 'Y', 2), (2, 'Y', 'N', 3), (3, 'Y', 'Y', 4), (4, 'N', 'N', 2)]
```

5. Declare a **dictionary** named **dict_no_of_burgers_sold** in main function to accumulate the quantity sold for each type of burger. It is more convenient to use the burger name as the key of this dictionary. You are NOT allowed to make the **dict_no_of_burgers_sold** variable global. The following diagrams show the detailed structure of **dict_no_of_burgers_sold**:

Dictionary **dict_no_of_burgers_sold**

Key	Data Type	Sample Value
Queen’s Signature	Integer	2
Cheeseburger	Integer	3
Chili Burger	Integer	4
Olive Burger	Integer	2
Beef Burger	Integer	3

Sample value of **dict_no_of_burgers_sold**

```
{'Queen's Signature': 2,
 'Cheeseburger': 3,
 'Chili Burger': 4,
 'Olive Burger': 2,
 'Beef Burger': 3
}
```

6. The main function should contain the following logics:
   - Read in burgers information from file named “**burger_list.csv**” located in the same folder.
   - Display a welcome message and followed by the main menu of Burger Queen Food Ordering system. Users can view all burgers information, buy different types of burgers and/or with combo set and cold drink, display sales statistics.

- To buy a burger, user is required to input burger type by using burger index 1-5, input Y/N to indicate whether a combo meal. User can input Y/N to indicate whether a cold drink is required if the user requires a combo meal, and input for quantity finally.
- User can buy multiple type of burgers in single order by inputting burger index again.
- To confirm order, user is required to input Enter in the burger menu without input of burger index.
- Display all Burger Queen shop statistics after order is confirmed. For details, please reference to sample output provided.
- Handle all errors inputted by user and **ask user to correct and input the same data again immediately**, such as but not limited to incorrect input of menu burger index, incorrect input string for combo meal and cold drink, incorrect input of quantity etc.

7. Write a function **compute_sales** which accepts four parameters (float for burger price, string Y/N for combo meal, string Y/N for cold drink, integer for burger quantity). This function does the following:
   - Get the selected burger's price, Y/N options for combo meal and cold drink, burger's quantity.
   - Use COMBO_MEAL_PRICE, COLD_DRINK_PRICE and burger's price to compute the amount of this order.

8. You may write any other reasonable variables and functions which might help you to complete this assignment.

Important Hints:

You may **implement some minor functions at last**, such as **reading burgers information from the file**, and most of the **input validation**. Such as validation of burger type, Y/N options for combo meal and cold drink, etc.

Instruction to students:

1. This is an End of Module Assessment and the weighting of this assignment is 20% of the Module Mark.

2. This assignment should be done by each individual student. Plagiarism will be treated seriously. All assignments that have been found involved wholly or partly in plagiarism (no matter these assignments are from the original authors or from the plagiarists) will score Zero marks.

3. You must use Python 3 to develop the programs.

4. Your programs must follow the style guide stated in PEP8 – Style Guide for Python Code published by python.org. https://www.python.org/dev/peps/pep-0008/. Marks may be deducted if the style guide is not followed.

Submission of students:

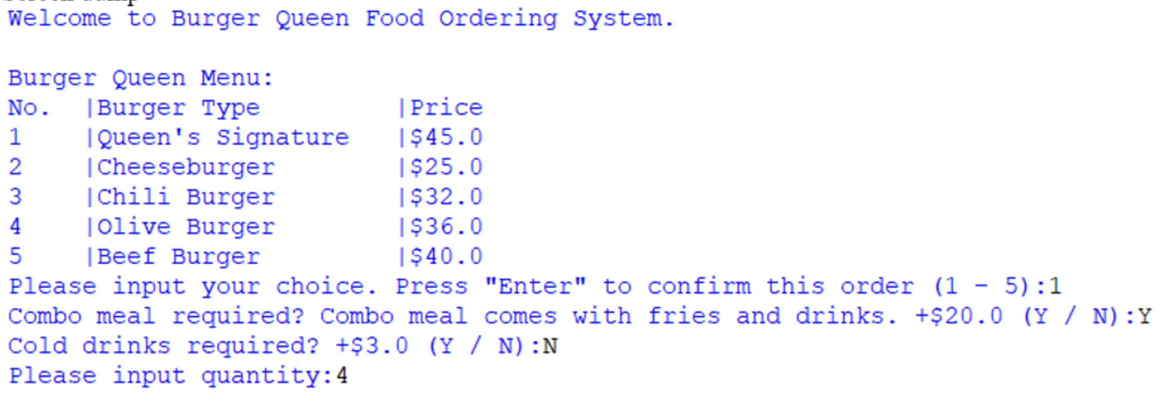
1. You are required to hand in:
   - Well-commented source code.
   - A test plan showing the evidence of testing.

2. Prepare a **word document** with a number of test cases showing different inputs for different situations that your program may encounter and how your program responses to show the capability of your program. **Your test cases should be enough to cover all invalid inputs** that could be provided by users.

3. For each test case, states the objective of the test case, input data and expected result. You should also include screen dump for each test run as evidence. Some samples of normal inputs and invalid inputs are provided.

<table>
<tr><th>ID</th><th>Test Case Name</th><th>Procedure</th><th>Expected Output</th><th>Result</th></tr>
<tr><td>1</td><td>Calculate the total price for 3 burger Queen's Signature with Combo meal option only</td><td>1. In the burger menu, input '1' for burger Queen's Signature
2. Enter 'Y' for combo meal required
3. Enter 'N' for cold drinks not required
4. Input '2' for quantity</td><td>The correct order "4 Queen's Signature(s) with combo set and hot drinks: $260.0" with total price of this order (i.e. $ 260) should be displayed</td><td>Pass / ~~Fail~~</td></tr>
<tr><td></td><td colspan="3">

Screen dump

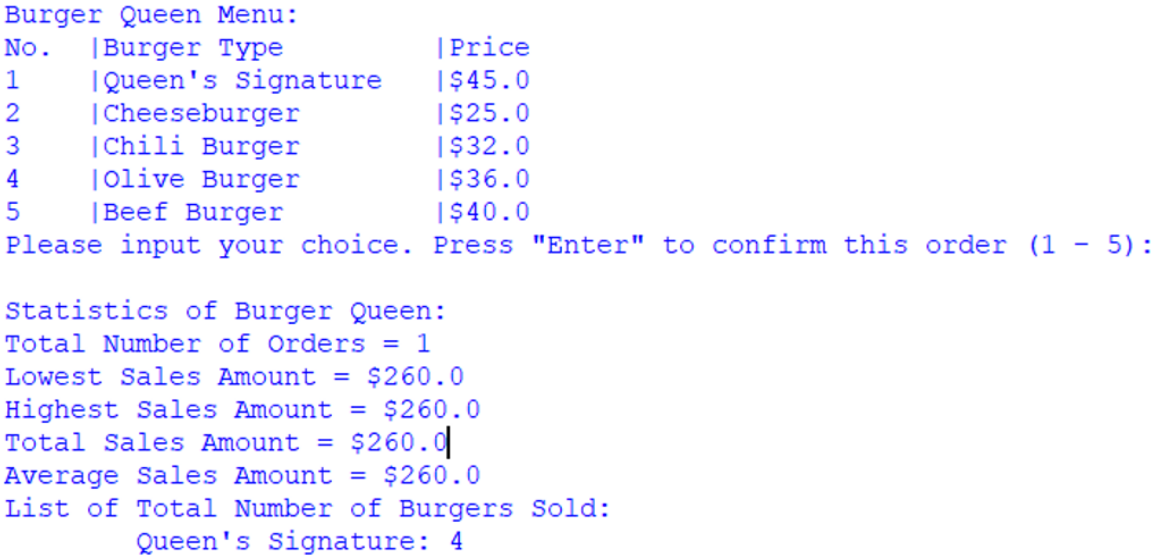


```
Welcome to Burger Queen Food Ordering System.

Burger Queen Menu:
No.  |Burger Type        |Price
1    |Queen's Signature  |$45.0
2    |Cheeseburger       |$25.0
3    |Chili Burger       |$32.0
4    |Olive Burger       |$36.0
5    |Beef Burger        |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):1
Combo meal required? Combo meal comes with fries and drinks. +$20.0 (Y / N):Y
Cold drinks required? +$3.0 (Y / N):N
Please input quantity:4
```

</td><td></td></tr>
<tr><td>2</td><td>Calculate the Statistics of Burger Queen after inputting the test case ID 1.</td><td>1. Press "Enter" in the burger menu after inputting the test case ID 1</td><td>The correct output should be displayed:
Total Number or Orders = 1
Lowest Sales Amount = $45
Highest Sales Amount = $45
Total Sales Amount = $45
Average Sales Amount = $45.0
List of Total Number of Burgers Sold:
Queen's Signature: 1</td><td>Pass / ~~Fail~~</td></tr>
<tr><td></td><td colspan="3">

Screen dump



```
Burger Queen Menu:
No.  |Burger Type        |Price
1    |Queen's Signature  |$45.0
2    |Cheeseburger       |$25.0
3    |Chili Burger       |$32.0
4    |Olive Burger       |$36.0
5    |Beef Burger        |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):

Statistics of Burger Queen:
Total Number of Orders = 1
Lowest Sales Amount = $260.0
Highest Sales Amount = $260.0
Total Sales Amount = $260.0
Average Sales Amount = $260.0
List of Total Number of Burgers Sold:
        Queen's Signature: 4
```

</td><td></td></tr>
<tr><td></td><td>…</td><td></td><td></td><td></td></tr>
</table>

5	Input invalid burger index	1. In the burger menu, input ‘0’	An error message “Invalid input for choice.” should be displayed and input for choice message should be shown again.	Pass / ~~Fail~~

Screen dump

```
Burger Queen Menu:
No.   |Burger Type         |Price
1     |Queen's Signature   |$45.0
2     |Cheeseburger        |$25.0
3     |Chili Burger        |$32.0
4     |Olive Burger        |$36.0
5     |Beef Burger         |$40.0
Please input your choice. Press "Enter" to confirm this order (1 - 5):0
Invalid input for choice
Please input your choice. Press "Enter" to confirm this order (1 - 5):
```

4. Submit all your works (in a zip file under the name of your student ID – e.g. 239999999.zip) to the Moodle website (http://moodle.vtc.edu.hk) by 11:55pm, 20 November 2023 (Monday). Late submission may score ZERO marks.

5. Each student will be required to conduct an assignment demonstration during laboratory class to show the system and walk through all the functions. The date of assignment demonstration will be scheduled during the lab session just after the assignment submission and before final practical test. Marks will be deducted if a student does not attend the assignment demonstration.

6. Marks Distribution
    - System Implementation (70%)
    - Validation on the input data and display appropriate error messages (20%)
    - Test plan (10%)