Debugging

- **Bug #1: Fixing Order Input Validation in get_order_input()** ***(Runtime Error)***
  - When entering a single value for the table number that is equal to or less than 10 the program would raise **TypeError**
    - Examples: “0” would raise **TypeError**, as well as entering “10”

Error Messages:

```
[dante@archbox temp]$ python main.py
--Main Menu--
1: Input order data
2: Change Menu Items
3: Finalize Order
------------------------------
Select an index: 1

Type order here: 0
Traceback (most recent call last):
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implemen
tation/temp/main.py", line 156, in <module>
    main()
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implemen
tation/temp/main.py", line 152, in main
    main_menu(menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implemen
tation/temp/main.py", line 123, in main_menu
    data = menu[str(index)][1](menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implemen
tation/temp/main.py", line 96, in get_order_input
    get_order(data, menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implemen
tation/temp/main.py", line 78, in get_order
    total_price, total_quantity, quantity_dict = get_totals(full_order)  # Cal
culates quantity and cost totals.
TypeError: cannot unpack non-iterable NoneType object
[dante@archbox temp]$
```

```
[dante@archbox temp]$ python main.py
--Main Menu--
1: Input order data
2: Change Menu Items
3: Finalize Order
------------------------------
Select an index: 1

Type order here: 10
Traceback (most recent call last):
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/main.py", line 156, in <module>
    main()
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/main.py", line 152, in main
    main_menu(menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/main.py", line 123, in main_menu
    data = menu[str(index)][1](menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/main.py", line 96, in get_order_input
    get_order(data, menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/main.py", line 78, in get_order
    total_price, total_quantity, quantity_dict = get_totals(full_order)  # Calculates quantity and cost totals.
TypeError: cannot unpack non-iterable NoneType object
[dante@archbox temp]$
```

Original Code:

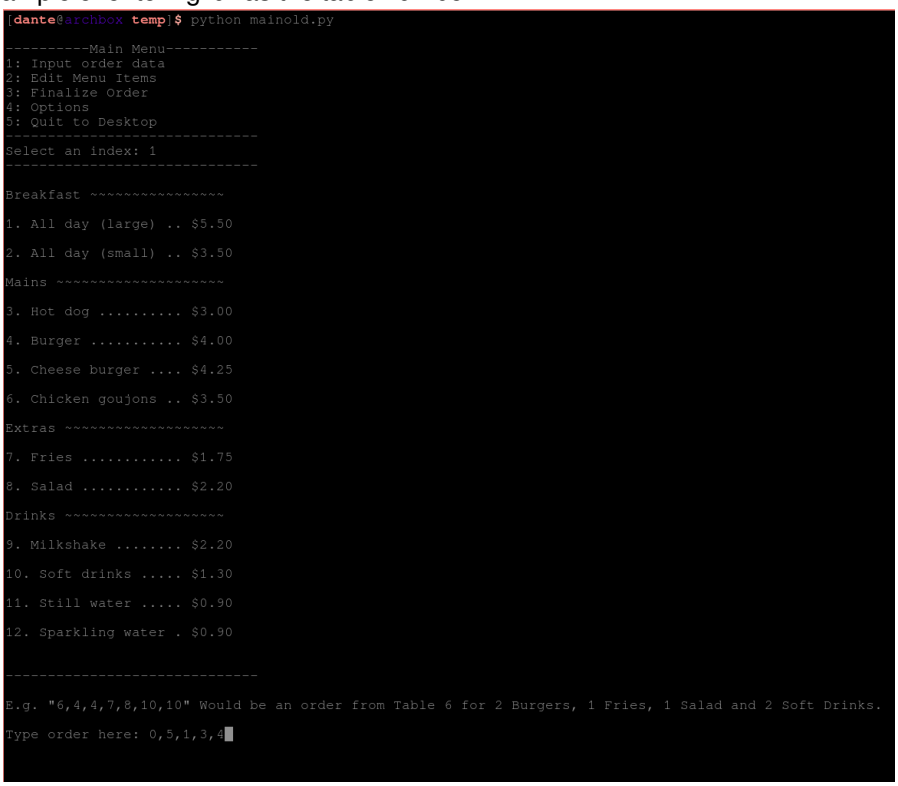
```
def get_order_input(menu_file):  # Validates Order
    while True:
        data = input("\nType order here: ").split(",")
        tmp_last_item = menu_file[len(menu_file) - 1]  # Assigns Last entry's index
        last_item = tmp_last_item[0]                   # to "last_item"
        invalid = ""
        for element in data:  # Takes each element in the array
            if len(element) > len(last_item) or element.isnumeric() == False:
                invalid = True  # ^If the digit length is greater than the digit length in the menu or is not integer
            elif int(element) > int(last_item):  # If the element is numerically greater than the last item
                invalid = True
        if invalid:  # If there are letters or symbols in the input:
            print("Your order has invalid characters, please try again.")
        else:
            break
    get_order(data, menu_file)
```

Fixed Code:

```
def get_order_input(menu_file):  # Validates Order
    while True:
        data = input("\nType order here: ").split(",")
        tmp_last_item = menu_file[len(menu_file) -  1]  # Assigns Last entry's index
        last_item = tmp_last_item[0]                    # to "last_item"
        invalid = ""
        invalidTable = ""
        for index, element in enumerate(data):  # Takes each element in the array
            if index == 0:  # If the element is first (table number doesn't apply for these rules)
                if element.isnumeric() == False:
                    invalid = True  # ^If the digit length is greater than 25 is not integer
                elif int(element) > 25:  # If the element is numerically greater than number of tables in (25)
                    invalidTable = True
            else:
                if len(element) > len(last_item) or element.isnumeric() == False:
                    invalid = True  # ^If the digit length is greater than the digit length in the menu or is not integer
                elif int(element) > int(last_item):  # If the element is numerically greater than the last item
                    invalid = True
        if invalidTable:
            print("We only have 25 tables! Table number must be lower than 25, please try again.")
        elif invalid:  # If there are letters or symbols in the input:
            print("Your order has invalid characters, please try again.")
        elif len(data) == 1:
            print("At least one order must be made per table, please try again.")
        else:
            break
    get_order(data, menu_file)
```

- How was the bug fixed?
  - The code will loop through each element in the user's input
  - The first element (the table number) will be checked to check to see:
    - IF: it is a numeric string
    - IF: The element is less than 25 *(at the time I was not aware that the maximum amount of tables was 10 and this was fixed later in the production of the program)*

- **Bug #2: Fixing Table number bug when entering "0" as a table number *(Logic Error)***
  - When entering the table number for the user's order as "0", the program would still accept it.
    - Example of entering"0" as the table number:



- The ouput shows order with the table number: “0”
- The program has accepted the table number when it should’ve rejected it instead because Table #0 doesn’t exist

```
------------------------------

E.g. "6,4,4,7,8,10,10" Would be an order from Table 6 for 2 Burgers, 1 Fries, 1 Salad and 2 Soft Drinks.

Type order here: 0,5,1,3,4

Your order for Table #0:

Cheese burger  ==  1
All day (large)  ==  1
Hot dog  ==  1
Burger  ==  1

Total Price = $16.75
Total Quantity of items ordered = 4

------------------------------



----------Main Menu-----------
1: Input order data
2: Edit Menu Items
3: Finalize Order
4: Options
5: Quit to Desktop
------------------------------
Select an index:
```

Fixed Code:

```
def get_order_input(menu_file):  # Validates Order Input from user
    print_menu(menu_file)
    while True:
        print("-" * 30)
        print("\nE.g. \"6,4,4,7,8,10,10\""
              " Would be an order from Table 6 for 2 Burgers, 1 Fries, 1 Salad and 2 Soft"
              " Drinks.")
        tmpData = input("\nType order here: ").split(",")

        # Removes spaces from each element in tmpData,
        # then adds elements to 'data'
        data = []
        for string in tmpData:  # Iterates over each value
            string = ''.join(string.split())  # Removes spaces from data
            data.append(string)

        tmp_last_item = menu_file[len(menu_file) -  1]  # Assigns Last entry's index
        last_item = tmp_last_item[0]                    # to "last_item"
        invalid = None
        invalidTable = None
        for index, element in enumerate(data):  # Takes each element in the array
            if index == 0:  # If the element is first (table number doesn't apply for these rules)
                if element.isnumeric() == False:  # If first element in data is NOT numeric:
                    invalid = True
                elif not 0 < int(element) <= 10:  # If the element is not numerically more than 0 and greater than number of tables (10)
                    invalidTable = True
            else:
                if len(element) > len(last_item) or element.isnumeric() == False:
                    invalid = True  # ^If the digit length is greater than the digit length in the menu or is not integer
                elif int(element) > int(last_item):  # If the element is numerically greater than the last item
                    invalid = True
        if invalidTable:
            print("\nWe only have 10 tables! Table number must be at least 1 and not more than 10, please try again.")
        elif invalid:  # If there are letters or symbols in the input:
            print("\nYour order has invalid characters, please try again.")
        elif len(data) == 1:
            print("\nAt least one order must be made per table, please try again.")
        else:
            break
    # data variable is input data in array format.
    total_price, total_quantity, quantity_dict, table_num = get_order(data, menu_file)
    return total_price, total_quantity, quantity_dict, table_num
```

- How was the bug fixed?
  - The code will loop through each element in the user’s input
  - The first element (the table number) will be checked to check to see:
    - IF it’s greater than 0 and less than or equal to 10. (Tables)
  - Originally the code would just check if it was less than 10 which would allow “0” to go through

- **Key:**
  - ***Green*** *= Additions made in the program*
  - ***Red*** *= Deletions in the program*

```
809  809
810  810          tmp_last_item = menu_file[len(menu_file) -  1]  # Assigns Last entry's index
811  811          last_item = tmp_last_item[0]                    # to "last_item"
812       -       invalid = ""
813       -       invalidTable = ""
     812  +       invalid = None
     813  +       invalidTable = None
814  814          for index, element in enumerate(data):  # Takes each element in the array
815  815              if index == 0:  # If the element is first (table number doesn't apply for these rules)
816  816                  if element.isnumeric() == False:  # If first element in data is NOT numeric:
817  817                      invalid = True
818       -              elif int(element) > 10:  # If the element is numerically greater than number of tables in (10)
     818  +              elif not 0 < int(element) <= 10:  # If the element is not numerically more than 0 and greater than number of tables (10)
819  819                      invalidTable = True
820  820              else:
821  821                  if len(element) > len(last_item) or element.isnumeric() == False:
822  822                      invalid = True  # ^If the digit length is greater than the digit length in the menu or is not integer
823  823                  elif int(element) > int(last_item):  # If the element is numerically greater than the last item
824  824                      invalid = True
825  825          if invalidTable:
826       -          print("\nWe only have 10 tables! Table number must be lower than 10, please try again.")
     826  +          print("\nWe only have 10 tables! Table number must be at least 1 and not more than 10, please try again.")
827  827          elif invalid:  # If there are letters or symbols in the input:
828  828              print("\nYour order has invalid characters, please try again.")
829  829          elif len(data) == 1:
```

- **Bug #3: Fixing Deletion of Menu Items Validation in delete_menu_items()** ***(Runtime Error)***
  - Upon entering a menu item index number that was greater than the maximum index number (out of range), the program would crash and raise **IndexError**
    - Examples: “” would raise **IndexError**, as well as entering “15”

Program Running with Error Message:

```
[dante@archbox temp]$ python mainold.py

----------Main Menu----------
1: Input order data
2: Edit Menu Items
3: Finalize Order
4: Options
5: Quit to Desktop
-----------------------------
Select an index: 2
-----------------------------

-------Edit Menu Items--------
1: Add menu items
2: Edit an existing menu item
3: Delete menu items
4: Exit to main menu
-----------------------------
Select an index: 3
```


```
Enter a menu item index to delete
Or type "E" to exit.

Your choice: 15
Traceback (most recent call last):
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/mainold.py", line 890, in <module>
    main_menu(menu_file)  # Main Menu for most the program
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/mainold.py", line 861, in main_menu
    menu_file = menu[str(index)][1](menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/mainold.py", line 597, in editing_main_menu
    menu_file = menu[str(index)][1](menu_file)
  File "/home/dante/personal/documents/github/NEA2021/95059_8188_FE_P/Implementation/temp/mainold.py", line 132, in delete_menu_items
    name = menu_file[delete_index-1][2]
IndexError: list index out of range
[dante@archbox temp]$
```

Original Code:

```
115
  1 # Allows user to delete menu items
  2 def delete_menu_items(menu_file):
  3     while True:
  4         print("-" * 30)
  5         print("\nThis is the current menu:")
  6         print_menu(menu_file)
  7
  8         # Asks the user what menu item index they want to delete
  9         print("Enter a menu item index to delete\n"
 10               "Or type \"E\" to exit.\n")
 11
 12         try:
 13             inp = input("Your choice: ")
 14             delete_index = int(inp)
 15
 16             # Assigns Last entry's index to "last_item"
 17             name = menu_file[delete_index-1][2]
 18             tmp_last_item = menu_file[len(menu_file) - 1]
 19             last_item_index = int(tmp_last_item[0])
 20             if delete_index > last_item_index or delete_index < 1:  # Out of range
 21                 print(f"Your delete index must be between: 1 and {last_item_index}\n"
 22                       "Please try again...\n")
 23             else:  # Index is valid
 24                 print(f"Are you sure you want to delete {name}?\n"
 25                       "- Enter \"Y\" to Confirm Menu Item Deletion\n"
 26                       "- Enter \"N\" to Discard Changes and Exit to Edit Menu\n")
 27
 28                 while True:  # Validates input for delete item or Exit
 29                     choiceInp = input("Your choice: ").lower()
 30
 31                     if choiceInp == "y":  # User wants to delete the menu item
 32                         # Deletes the menu item
 33                         del menu_file[delete_index-1]
 34
 35                         # Corrects menu index numbers in the list
 36                         for index, full_item in enumerate(menu_file, start=1):
 37                             full_item[0] = str(index)
 38
 39                         print(f"{name} Deleted!")
 40                         break
 41
 42                     elif choiceInp == "n":  # User wants to exit
 43                         break
 44                     else:
 45                         print("Please enter \"y\" or \"n\" as your choice, try again.\n")
 46
 47                 if choiceInp == "n":  # User exits instead of deleting menu item
 48                     print("Exiting...\n")
 49                     break
 50
 51         except ValueError:  # User entered input other than an integer
 52             if inp == "e":
 53                 break
 54             else:
 55                 print("Please enter numeric characters only, try again.\n")
 56     return menu_file
 57
NORMAL  mainold.py
```

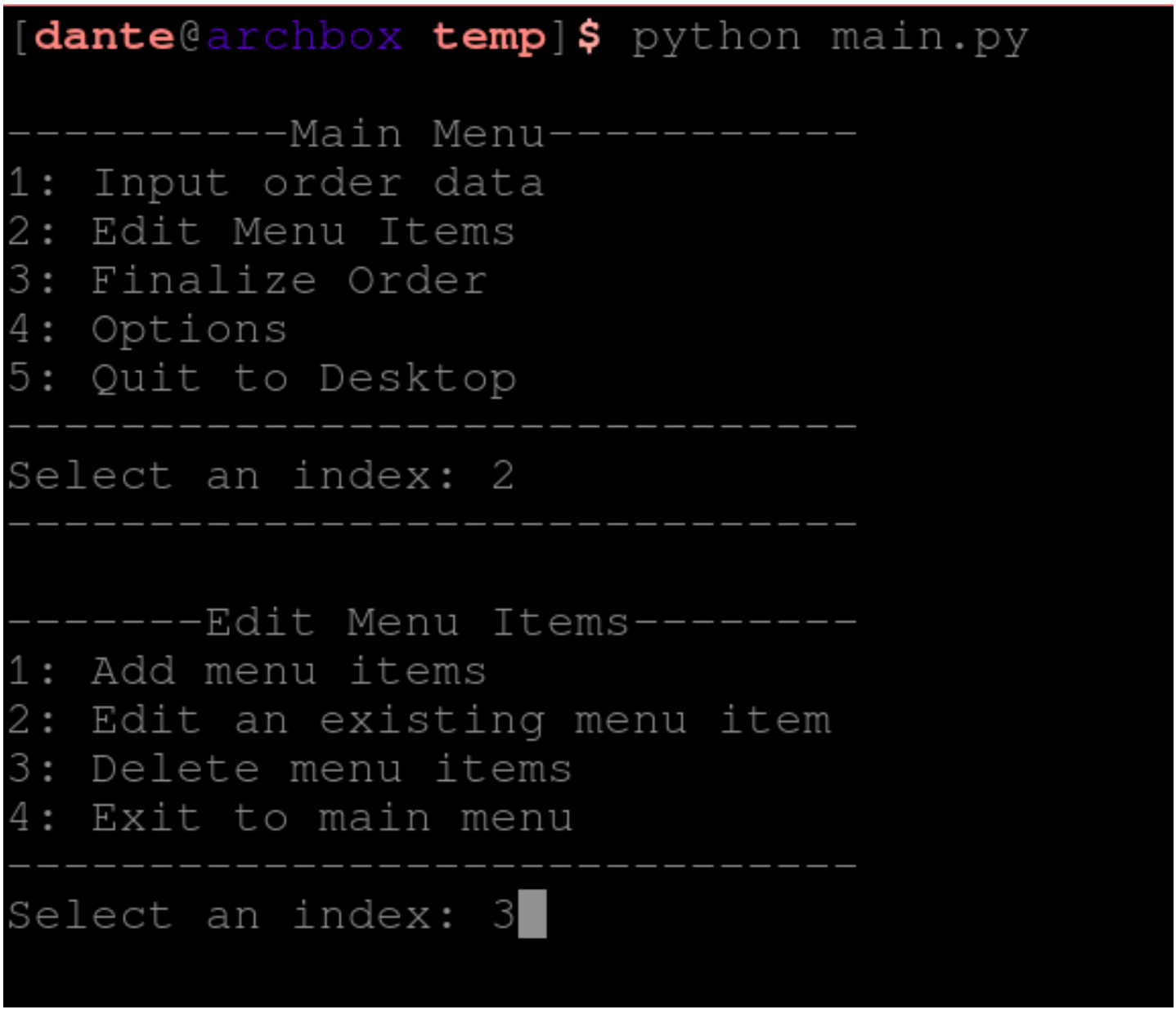
Fixed Code:

```
115
  1 # Allows user to delete menu items
  2 def delete_menu_items(menu_file):
  3     while True:
  4         print("-" * 30)
  5         print("\nThis is the current menu:")
  6         print_menu(menu_file)
  7
  8         # Asks the user what menu item index they want to delete
  9         print("Enter a menu item index to delete\n"
 10               "Or type \"E\" to exit.\n")
 11
 12         while True:  # Checks if user input is in range for deletion or if the user wants to exit "delete_menu_items"
 13             try:
 14                 inp = input("Your choice: ")
 15                 int_inp = int(inp)  # Converts to type: integer
 16
 17                 tmp_last_item = menu_file[len(menu_file) - 1]
 18                 last_item_index = int(tmp_last_item[0])
 19                 delete_index = int_inp
 20                 if delete_index > last_item_index or delete_index < 1:  # Out of range
 21                     print(f"Your delete index must be between: 1 and {last_item_index}\n"
 22                           "Please try again...\n")
 23                 else:
 24                     break
 25             except ValueError:  # Input is not an integer
 26                 if inp == "e":
 27                     break
 28                 else:
 29                     print("Please enter numeric characters only, try again\n")
 30
 31         if inp == "e":
 32             break
 33
 34         # Assigns Last entry's index to "last_item"
 35         name = menu_file[delete_index-1][2]
 36
 37         print(f"Are you sure you want to delete {name}?\n"
 38               "- Enter \"Y\" to Confirm Menu Item Deletion\n"
 39               "- Enter \"N\" to Discard Changes and Exit to Edit Menu\n")
 40
 41         while True:  # Validates input for delete item or Exit
 42             choiceInp = input("Your choice: ").lower()
 43
 44             if choiceInp == "y":  # User wants to delete the menu item
 45                 # Deletes the menu item
 46                 del menu_file[delete_index-1]
 47
 48                 # Corrects menu index numbers in the list
 49                 for index, full_item in enumerate(menu_file, start=1):
 50                     full_item[0] = str(index)
 51
 52                 print(f"{name} Deleted!")
 53                 break
 54
 55             elif choiceInp == "n":  # User wants to exit
 56                 break
 57             else:
 58                 print("Please enter \"y\" or \"n\" as your choice, try again.\n")
 59
 60         if choiceInp == "n":  # User exits instead of deleting menu item
 61             print("Exiting...\n")
 62             break
 63
 64
 65     return menu_file
NORMAL  main.py
```

- How was the bug fixed?
  - The program will initialize the variable *last_item_index* (the menu item in the menu with the greatest index number)
  - Then the program will evaluate if the user's input is greater than value of *last_item_index.*
    - IF the delete_index (deletion index that the user entered) is greater than the *last_item_index* less than 1 THEN: ***Ask to enter another value again.***
  - *Originally the code would raise **IndexError** upon the user entering an invalid index because the user specified index doesn't exist.*
  - **Key:**
    - ***Green** = Additions made in the program*
    - ***Red** = Deletions in the program*

```
            print("Enter a menu item index to delete\n"
                  "Or type \"E\" to exit.\n")

        try:
            inp = input("Your choice: ")
            delete_index = int(inp)

            # Assigns last entry's index to "last_item"
            name = menu_file[delete_index-1][2]
            tmp_last_item = menu_file[len(menu_file) - 1]
            last_item_index = int(tmp_last_item[0])
            if delete_index > last_item_index or delete_index < 1:  # Out of range
                print(f"Your delete index must be between: 1 and {last_item_index}\n"
                      "Please try again...\n")
            else:  # Index is valid
                print(f"Are you sure you want to delete {name}?\n"
                      "- Enter \"Y\" to Confirm Menu Item Deletion\n"
                      "- Enter \"N\" to Discard Changes and Exit to Edit Menu\n")

                while True:  # Validates input for delete item or Exit
                    choiceInp = input("Your choice: ").lower()
        while True:  # Checks if user input is in range for deletion or if the user wants to exit "delete_menu_items"
            try:
                inp = input("Your choice: ")
                int_inp = int(inp)  # Converts to type: integer

                    if choiceInp == "y":  # User wants to delete the menu item
                        # Deletes the menu item
                        del menu_file[delete_index-1]
                tmp_last_item = menu_file[len(menu_file) - 1]
                last_item_index = int(tmp_last_item[0])
                delete_index = int_inp
                if delete_index > last_item_index or delete_index < 1:  # Out of range
                    print(f"Your delete index must be between: 1 and {last_item_index}\n"
                          "Please try again...\n")
                else:
                    break
            except ValueError:  # Input is not an integer
                if inp == "e":
                    break
                else:
                    print("Please enter numeric characters only, try again\n")

                    # Corrects menu index numbers in the list
                    for index, full_item in enumerate(menu_file, start=1):
                        full_item[0] = str(index)
        if inp == "e":
            break

                    print(f"{name} Deleted!")
                    break
        # Assigns last entry's index to "last_item"
        name = menu_file[delete_index-1][2]

                    elif choiceInp == "n":  # User wants to exit
                        break
                    else:
                        print("Please enter \"y\" or \"n\" as your choice, try again.\n")
        print(f"Are you sure you want to delete {name}?\n"
              "- Enter \"Y\" to Confirm Menu Item Deletion\n"
              "- Enter \"N\" to Discard Changes and Exit to Edit Menu\n")

                if choiceInp == "n":  # User exits instead of deleting menu item
                    print("Exiting...\n")
                    break
        while True:  # Validates input for delete item or Exit
            choiceInp = input("Your choice: ").lower()

            if choiceInp == "y":  # User wants to delete the menu item
                # Deletes the menu item
                del menu_file[delete_index-1]

        except ValueError:  # User entered input other than an integer
            if inp == "e":
                # Corrects menu index numbers in the list
                for index, full_item in enumerate(menu_file, start=1):
                    full_item[0] = str(index)

                print(f"{name} Deleted!")
                break

            elif choiceInp == "n":  # User wants to exit
                break
            else:
                print("Please enter numeric characters only, try again.\n")
                print("Please enter \"y\" or \"n\" as your choice, try again.\n")

        if choiceInp == "n":  # User exits instead of deleting menu item
            print("Exiting...\n")
            break


    return menu_file
```

Fixed Code Running without Errors:



```
------------------------------
------------------------------

This is the current menu:

Breakfast ~~~~~~~~~~~~~~~~~

1. All day (large) .. $5.50

2. All day (small) .. $3.50

Mains ~~~~~~~~~~~~~~~~~~~~~

3. Hot dog ......... $3.00

4. Burger .......... $4.00

5. Cheese burger .... $4.25

6. Chicken goujons .. $3.50

Extras ~~~~~~~~~~~~~~~~~~~~

7. Fries ........... $1.75

8. Salad ........... $2.20

Drinks ~~~~~~~~~~~~~~~~~~~~

9. Milkshake ........ $2.20

10. Soft drinks ..... $1.30

11. Still water ..... $0.90

12. Sparkling water . $0.90


Enter a menu item index to delete
Or type "E" to exit.

Your choice: 15
```

```
4. Burger ..... $4.00

5. Cheese burger .... $4.25

6. Chicken goujons .. $3.50

Extras ~~~~~~~~~~~~~~~~~~

7. Fries ..... $1.75

8. Salad ..... $2.20

Drinks ~~~~~~~~~~~~~~~~~~

9. Milkshake ........ $2.20

10. Soft drinks ..... $1.30

11. Still water ..... $0.90

12. Sparkling water . $0.90


Enter a menu item index to delete
Or type "E" to exit.

Your choice: 15
Your delete index must be between: 1 and 12
Please try again...

Your choice: 1000099
Your delete index must be between: 1 and 12
Please try again...

Your choice: fooBar
Please enter numeric characters only, try again

Your choice: !!!!!
Please enter numeric characters only, try again

Your choice:
```