



Document tabs

Tab 1

Headings you add to the document will appear here.

Sabanci University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Summer 2024-2025

Take Home Exam 3 – Starbucks Order Management System
Due: Monday, 18 August 2025, 23:55 (SHARP)

DISCLAIMER:

Only checking the sample run cases might not be sufficient as your solution will be checked against a variety of samples different from the provided samples; however checking these cases is highly encouraged and recommended.

You can NOT collaborate with your friends and discuss your solutions with each other. You have to write down the code on your own. Plagiarism will not be tolerated AND cooperation is not an excuse!

Introduction

The aim of this THE assignment is to practice dynamic queue and stack data structures implemented via linked lists. In this THE, you will build an order management system using a Dynamic Queue for orders and Dynamic Stacks for drinks within each order, building on the Circular Doubly Linked List menu from THE2. Imagine you are continuing as the senior programmer at Starbucks' headquarters in Istanbul. Following the positive customer feedback on your online menu system, the Starbucks main branch boss has decided to entrust you with developing an order management system to efficiently register and handle Starbucks orders. Given the manager's familiarity with queue and stack concepts, he has specifically required you to design the order structure using a Dynamic Queue and Dynamic Stacks, which must be implemented as classes with proper destructors. He expects you to first establish the core infrastructure of the system and then implement the necessary functions on top of it.

You should be careful to fully adhere to using just the mentioned data structures in your code, and avoid incorporating any miscellaneous data structures that are not requested. This is crucial to earning full credit.

Steps to Build a Comprehensive Menu Manager Program

Tab 1

Headings you add to the document will appear here.

Your task is to develop a basic Starbucks order placement system that retrieves menu items from the previously built data structure (Circular Doubly Linked List - CDLL) created in THE2, solely to preserve the menu items. Please note that we will not regrade the functions from THE2. They are included here only for loading menu items and using them to place orders, by reading the menu text file and storing it in the CDLL. Subsequently, as each customer arrives and places an order, it must be registered in the dynamic queue, which is implemented using linked list logic. A customer may include as many drinks as desired in a single order, but the process of delivering the items in an order to the customer and retrieving each item must be handled using the dynamic stack logic.

Note: Be careful. Implement the program in an organized manner with modular functions. Use meaningful identifiers, proper indentation, and comments where appropriate. This assignment emphasizes pointer management, memory deallocation, and the correct use of queue/stack operations. Errors in these areas will result in grade deductions. There will be significant grade reductions if the queue and stack are not implemented as linked lists, and a direct zero if any other container data structures are used (including built-in arrays, vectors, dynamic arrays, and similar). Additionally, you can expect substantial penalties for missing or improper deallocation, as memory leaks will be checked.

Roadmap

Format of Drink Items

Here we introduce and describe the format of each order.

Each order contains an order ID (automatically assigned by the program, starting from number 100 and automatically increasing by 10 after placing another order) and details for sugar packs added to individual drinks within the order.

You can see the overall structure of the ordering system in the below simplified sketch.

Tab 1

Headings you add to the document will appear here.

will appear here.

Dynamic Queue



As shown in the sketch, each order consists of an order ID (auto-incremented by the program), a dynamic stack of drink details, and the total order price.

We also provide a quick look at the structure of each individual drink item in a single order:



Each drink item in an order contains two details. The drink name and the number of sugar packs dissolved in that drink.

In this order management system, we have multiple necessary functions that will directly work with orders within the dynamic queue.

As stated, you are expected to utilize only linked list structures to implement the dynamic queue and the dynamic stack used for the system. Any other approach will result in a grade of 0 for the entire assignment automatically.

Functions

In this section, you are expected to build some functions that correspond to the menu options. These must be implemented using dynamic queue and dynamic stack via linked lists, with careful attention to pointer operations and memory management. *Keep in mind that, beyond taking and preparing orders, the administrator requires constant access to up-to-date statistics on total drinks sold, total sugar packs used, total revenue, and the number of completed orders.* Therefore, maintain these metrics by updating them during order preparation for later reference through the report option. Below, we provide a detailed overview of the expected behavior and general performance for each option you need to implement in this THE. These options correspond to the menu choices and must be handled in your program. Pay close attention to proper memory handling in operations that add or remove items to avoid issues, and ensure all cases are covered as described. Additionally, carefully consider the input parsing and validation to match the specified formats.

The menu options are.

1. Display today's Starbucks menu
2. Place a new order
3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)

1. Display Today's Starbucks Menu (Choice "1")

This option lets you view the full list of drinks available in the menu by traversing the circular doubly linked list in forward order, showing all details for each item.

- **Inputs.** None additional. The menu is already loaded at program start.
- **Outputs.** For each drink in the list. "ID: [id]\nName: [name]\nRating: [rating]\nPrice: [price]\n---\n", where id is an integer, name is a string, rating and price are double values. If no items exist, "The menu is empty.\n---\n".
- **Notes.** The display starts from the first node and loops until back to start. Rating and price are shown as entered from the file. This does not modify the menu.

Tab 1

Headings you add to the document will appear here.

Tab 1

Headings you add to the document will appear here.

Tab 1

Headings you add to the document will appear here.

Useful for checking available drinks before ordering.

[2. Place a New Order \(Choice "2"\)](#)

This option enables the creation of a new customer order by prompting for multiple drink entries, validating each against the menu, accumulating the total cost, and adding the order to the end of the queue with an auto-generated ID.

- **Inputs.** The program outputs "Enter drink details (name and sugar) one by one. Type 'enough' to stop.\n", followed by repeated "Enter: " prompts. Each drink input is a line with string name and integer sugar separated by space. Entry stops on "enough" (case-insensitive).
- **Outputs.** If valid drinks were added, "Order No. [id] added to the queue.\n---\n" where id starts at 100 and increases by 10. If no valid drinks were added at all, i.e. drink not available in the menu, "No drinks added.\n---\n". For invalid name, "Drink '[name]' not found. Try again.\n". For bad format, "Invalid format. Enter name and sugar.\n".

Tab 1

Headings you add to the document will appear here.

- **Notes.** Name comparison ignores case. Total price sums menu prices of valid drinks. Sugar packs are integers (assumed non-negative) and they don't have any effect on the total price. Invalid lines prompt again. Empty orders are not added. Drinks are stored in LIFO for later retrieval.

[3. Prepare the Next Order \(Choice "3"\)](#)

This option processes the order at the front of the queue by removing it, displaying its details including all drinks in reverse addition order, and updating cumulative statistics with the order's data.

- **Inputs.** None.
- **Outputs.** "Preparing Order No. [id]\nTotal price: [price]\nDrinks (LIFO):\n[drinkName] with [sugarPacks] sugar packs\n...\n---\n". If the queue is empty, "No orders to prepare.\n---\n".
- **Notes.** Price is displayed as double. Drinks are listed starting from the most recently added. Updates include incrementing completed orders count, adding to total drinks sold, summing sugar packs used, and adding to total revenue. Resources for the order are released after processing.

[4. Display Current Orders in Preparation Queue \(Choice "4"\)](#)

This option provides a complete view of all orders currently in the queue, detailing their IDs, full drink lists in reverse addition order, and prices, without any modifications.

- **Inputs.** None.
- **Outputs.** "Current orders:\nOrder No. [id]\nDrinks:\n[drinkName] with [sugarPacks] sugar packs\n...\nTotal Price: [price]\n---\n" for each order from front to rear. If empty, "No orders in the queue.\n---\n".
- **Notes.** Orders are shown in their current sequence following queue logic. Drinks per order start from the most recent one using stack logic. Prices as doubles. This view helps verify the queue before actions like prepare or rush. The logic behind the queue and stack should never be broken.

[5. View Starbucks Order Report \(Choice "5"\)](#)

This option summarizes the key performance metrics accumulated from all previously processed orders, giving an overview of operations so far.

- **Inputs.** None.
- **Outputs.** "Starbucks Order Report:\nCompleted orders: [completedOrders]\nDrinks sold: [totalDrinksSold]\nSugar packs: [totalSugarPacks]\nRevenue: [totalRevenue]\n---\n", with counts as integers and revenue as double.

- **Notes.** Reflects only prepared orders. Revenue includes any price adjustments from rushes. Can be called anytime to see current stats. No reset mechanism.

[6. Cancel an Order \(Choice "6"\)](#)

This option locates a specific order by ID and removes it from the queue, discarding its details without affecting statistics.

- **Inputs.** "Enter Order No. to cancel: " expecting an integer. If input invalid, "Invalid input.\n---\n".
- **Outputs.** "Order No. [id] canceled.\n---\n" if removed. "Order No. [id] not

Tab 1

Headings you add to the document will appear here.

Tab 1

⋮

Headings you add to the document will appear here.

found.\n---\n" if absent. "No orders to cancel.\n---\n" if the queue empty.

- **Notes.** ID must match exactly. During removal queue logic should be preserved. Removal handles positions like front or rear. Drink details are cleared. The queue may be empty after cancellation.

7. Rush an Order (Choice "7")

This option elevates a specified order to priority by repositioning it to the front of the queue and increasing its total price by 30 percent.

- **Inputs.** "Enter Order No. to rush: " expecting an integer. If input invalid, "Invalid input.\n---\n".
- **Outputs.** "Order No. [id] moved to front and rushed (price +30%).\n---\n" if moved. "Order No. [id] rushed (price +30%).\n---\n" if no move. "Order No. [id] not found.\n---\n" if absent. "No orders to rush.\n---\n" if empty.
- **Notes.** New price is original * 1.3, affecting future displays and revenue. Repositioning adjusts links. If at rear, queue updates accordingly. ID exact match is required.

8. Exit (prepare all remaining orders first) (Choice "8")

This option ensures all pending orders are handled by processing them sequentially before shutting down the system and cleaning up resources.

- **Inputs.** None.
- **Outputs.** "Preparing remaining orders...\n" followed by preparing outputs for each order (or empty message if none). Then "Exiting...". "No orders left to prepare.\nExiting..." if the queue is empty.

Tab 1

⋮

Headings you add to the document will appear here.

- **Notes.** Call preparation repeatedly until the queue clears. Then releases menu CDLL nodes via traversal. Ends the input loop. Final stats include these preparations.

Sample Runs

In this section, we provide sample runs for each primary function of the program, covering all expected behaviors and edge cases. You must carefully review these sample runs, as your program is required to produce identical outputs and handle all scenarios shown, including edge cases, to meet the assignment requirements.

Before proceeding to the sample runs, we list the items in MondayMenu.txt used in tests

MondayMenu.txt

Cappuccino 4.3 150
Flat-White 4.4 155
Mocha 4.6 180
Chai-Latte 4.2 170
Iced-Coffee 4.0 145
Macchiato 4.1 140
Turkish-Coffee 4.7 130
Caramel-Macchiato 4.5 175
Matcha-Latte 4.4 190
Hot-Chocolate 4.8 165

Note: user inputs are in **Orange**, terminal outputs (text) are in **Blue**

Tab 1

⋮

Headings you add to the document will appear here.

1. Display today's Starbucks menu

Enter menu filename (e.g., menu.txt): **MondayMenu.txt**
The drink Cappuccino is added to the menu.
The drink Flat-White is added to the menu.
The drink Mocha is added to the menu.
The drink Chai-Latte is added to the menu.
The drink Iced-Coffee is added to the menu.
The drink Macchiato is added to the menu.
The drink Turkish-Coffee is added to the menu.
The drink Caramel-Macchiato is added to the menu.
The drink Matcha-Latte is added to the menu.
The drink Hot-Chocolate is added to the menu.

Starbucks Order Management System

Tab 1

⋮

Headings you add to the document will appear here.

Options:

1. Display today's Starbucks menu
2. Place a new order

```

3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)
---
Enter choice (1-8): 1
---
Invalid choice (1-8).
---
Enter choice (1-8): 1
Please note that after providing input, pressing the Enter key adds a newline. In your program, ensure that this newline is handled appropriately and avoid adding extra '\n' or 'endl' when processing cin input.
---
ID: 1
Name: Cappuccino
Rating: 4.3
Price: 150
---
ID: 2
Name: Flat-White
Rating: 4.4
Price: 155
---
ID: 3
Name: Mocha
Rating: 4.6
Price: 180
---
ID: 4
Name: Chai-Latte
Rating: 4.2
Price: 170
---
ID: 5
Name: Iced-Coffee
Rating: 4
Price: 145
---
ID: 6
Name: Macchiato
Rating: 4.1
Price: 140
---
ID: 7
Name: Turkish-Coffee
Rating: 4.7

```

Tab 1

Headings you add to the document will appear here.

```

Price: 130
---
ID: 8
Name: Caramel-Macchiato
Rating: 4.5
Price: 175
---
ID: 9
Name: Matcha-Latte
Rating: 4.4
Price: 190
---
ID: 10
Name: Hot-Chocolate
Rating: 4.8
Price: 165
---
Enter choice (1-8): 8
---
No orders left to prepare.
Exiting...

```

2. Place a new order

```

Enter menu filename (e.g., menu.txt): MondayMenu.txt
The drink Cappuccino is added to the menu.
The drink Flat-White is added to the menu.
The drink Mocha is added to the menu.
The drink Chai-Latte is added to the menu.
The drink Iced-Coffee is added to the menu.
The drink Macchiato is added to the menu.
The drink Turkish-Coffee is added to the menu.
The drink Caramel-Macchiato is added to the menu.
The drink Matcha-Latte is added to the menu.
The drink Hot-Chocolate is added to the menu.
---
Starbucks Order Management System
---
Options:
1. Display today's Starbucks menu
2. Place a new order
3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)
---
Enter choice (1-8): 2

```

Tab 1

Headings you add to the document will appear here.

Tab 1

Headings you add to the document will appear here.

```
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: turkish-coffee 3  
Enter: mocha 5  
Enter: cold-chocolate 12  
Drink 'cold-chocolate' not found. Try again.  
Enter: test-drink what 383  
Invalid format. Enter name and sugar.  
Enter: Chai-Latte 0  
Enter: enough  
Order No. 100 added to the queue.  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: enough  
No drinks added.  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: xyz-xyz seven  
Invalid format. Enter name and sugar.  
Enter: Nothing-drink 666  
Drink 'Nothing-drink' not found. Try again.  
Enter: enough  
No drinks added.  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: Flat-White 1  
Enter: enough  
Order No. 110 added to the queue.  
---  
Enter choice (1-8): 8  
---  
Preparing remaining orders...  
Preparing Order No. 100  
Total price: 480  
Drinks (LIFO):  
Chai-Latte with 0 sugar packs  
mocha with 5 sugar packs  
turkish-coffee with 3 sugar packs  
---  
Preparing Order No. 110  
Total price: 155  
Drinks (LIFO):  
Flat-White with 1 sugar packs
```

Tab 1

Headings you add to the document will appear here.

```
---  
Exiting...  


---



### 3. Prepare the next order



```
Enter menu filename (e.g., menu.txt): MondayMenu.txt
The drink Cappuccino is added to the menu.
The drink Flat-White is added to the menu.
The drink Mocha is added to the menu.
The drink Chai-Latte is added to the menu.
The drink Iced-Coffee is added to the menu.
The drink Macchiato is added to the menu.
The drink Turkish-Coffee is added to the menu.
The drink Caramel-Macchiato is added to the menu.
The drink Matcha-Latte is added to the menu.
The drink Hot-Chocolate is added to the menu.

Starbucks Order Management System

Options:
1. Display today's Starbucks menu
2. Place a new order
3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)

Enter choice (1-8): 3

No orders to prepare.

Enter choice (1-8): 2

Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Chai-Latte 1
Enter: Cappuccino 0
Enter: Hot-Chocolate 12
Enter: enough
Order No. 100 added to the queue.
```


```

Tab 1

Headings you add to the document will appear here.

will appear here.

```
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: Chai-Latte 12  
Enter: Chai-Latte 8  
Enter: Chai-Latte 1  
Enter: enough
```

Tab 1

Headings you add to the document will appear here.

```
Order No. 110 added to the queue.  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: Chai-Latte 12  
Enter: Chai 13  
Drink 'Chai' not found. Try again.  
Enter: turkish-coffee 0  
Enter: enough  
Order No. 120 added to the queue.  
---  
Enter choice (1-8): 3  
---  
Preparing Order No. 100  
Total price: 485  
Drinks (LIFO):  
Hot-Chocolate with 12 sugar packs  
Cappuccino with 0 sugar packs  
Chai-Latte with 1 sugar packs  
---  
Enter choice (1-8): 3  
---  
Preparing Order No. 110  
Total price: 510  
Drinks (LIFO):  
Chai-Latte with 1 sugar packs  
Chai-Latte with 8 sugar packs  
Chai-Latte with 12 sugar packs  
---  
Enter choice (1-8): 8  
---  
Preparing remaining orders...  
Preparing Order No. 120  
Total price: 300  
Drinks (LIFO):  
turkish-coffee with 0 sugar packs  
Chai-Latte with 12 sugar packs  
---  
Exiting...
```

Tab 1

Headings you add to the document will appear here.

4. Display current orders in preparation queue

```
Enter menu filename (e.g., menu.txt): MondayMenu.txt  
The drink Cappuccino is added to the menu.  
The drink Flat-White is added to the menu.  
The drink Mocha is added to the menu.  
The drink Chai-Latte is added to the menu.  
The drink Iced-Coffee is added to the menu.
```

Tab 1

Headings you add to the document will appear here.

```
The drink Macchiato is added to the menu.  
The drink Turkish-Coffee is added to the menu.  
The drink Caramel-Macchiato is added to the menu.  
The drink Matcha-Latte is added to the menu.  
The drink Hot-Chocolate is added to the menu.  
---  
Starbucks Order Management System  
---  
Options:  
1. Display today's Starbucks menu  
2. Place a new order  
3. Prepare the next order  
4. Display current orders in preparation queue  
5. View Starbucks order report  
6. Cancel an order  
7. Rush an order  
8. Exit (prepare all remaining orders first)  
---  
Enter choice (1-8): 4  
---  
No orders in the queue.  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: Caramel-Macchiato 12  
Enter:  
Invalid format. Enter name and sugar.  
Enter: Cappuccino 1
```

```

...
Enter: Cappuccino 1
Enter: enough
Order No. 100 added to the queue.
---
Enter choice (1-8): 4
---
Current orders:
Order No. 100
Drinks:
Cappuccino with 1 sugar packs
Cappuccino with 1 sugar packs
Caramel-Macchiato with 12 sugar packs
Total Price: 475
---
Enter choice (1-8): 2
---
Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Flat-White 7
Enter: Iced-Coffee 10
Enter: Hot-Chocolate 0

```

Tab 1

Headings you add to the document will appear here.

```

Enter: enough
Order No. 110 added to the queue.
---
Enter choice (1-8): 4
---
Current orders:
Order No. 100
Drinks:
Cappuccino with 1 sugar packs
Cappuccino with 1 sugar packs
Caramel-Macchiato with 12 sugar packs
Total Price: 475
---
Order No. 110
Drinks:
Hot-Chocolate with 0 sugar packs
Iced-Coffee with 10 sugar packs
Flat-White with 7 sugar packs
Total Price: 465
---
Enter choice (1-8): 8
---
Preparing remaining orders...
Preparing Order No. 100
Total price: 475
Drinks (LIFO):
Cappuccino with 1 sugar packs
Cappuccino with 1 sugar packs
Caramel-Macchiato with 12 sugar packs
---
Preparing Order No. 110
Total price: 465
Drinks (LIFO):
Hot-Chocolate with 0 sugar packs
Iced-Coffee with 10 sugar packs
Flat-White with 7 sugar packs
---
Exiting...

```

5. View Starbucks order report

```

Enter menu filename (e.g., menu.txt): MondayMenu.txt
The drink Cappuccino is added to the menu.
The drink Flat-White is added to the menu.
The drink Mocha is added to the menu.
The drink Chai-Latte is added to the menu.
The drink Iced-Coffee is added to the menu.
The drink Macchiato is added to the menu.
The drink Turkish-Coffee is added to the menu.

```

Tab 1

Headings you add to the document will appear here.

```

The drink Caramel-Macchiato is added to the menu.
The drink Matcha-Latte is added to the menu.
The drink Hot-Chocolate is added to the menu.
---
Starbucks Order Management System
---
Options:
1. Display today's Starbucks menu
2. Place a new order
3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)
---
Enter choice (1-8): 5

```

Tab 1

Headings you add to the document will appear here.

```
---  
Starbucks Order Report:  
Completed orders: 0  
Drinks sold: 0  
Sugar packs: 0  
Revenue: 0  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: Turkish-Coffee 10  
Enter: Flat-White 5  
Enter: Hot-Chocolate 0  
Enter: enough  
Order No. 100 added to the queue.  
---  
Enter choice (1-8): 2  
---  
Enter drink details (name sugar) one by one. Type 'enough' to stop.  
Enter: Mocha 5  
Enter: Matcha-Latte 20  
Enter: enough  
Order No. 110 added to the queue.  
---  
Enter choice (1-8): 5  
---  
Starbucks Order Report:  
Completed orders: 0  
Drinks sold: 0  
Sugar packs: 0  
Revenue: 0  
---
```

Tab 1

Headings you add to the document will appear here.

```
Enter choice (1-8): 3  
---  
Preparing Order No. 100  
Total price: 450  
Drinks (LIFO):  
Hot-Chocolate with 0 sugar packs  
Flat-White with 5 sugar packs  
Turkish-Coffee with 10 sugar packs  
---  
Enter choice (1-8): 5  
---  
Starbucks Order Report:  
Completed orders: 1  
Drinks sold: 3  
Sugar packs: 15  
Revenue: 450  
---  
Enter choice (1-8): 3  
---  
Preparing Order No. 110  
Total price: 370  
Drinks (LIFO):  
Matcha-Latte with 20 sugar packs  
Mocha with 5 sugar packs  
---  
Enter choice (1-8): 5  
---  
Starbucks Order Report:  
Completed orders: 2  
Drinks sold: 5  
Sugar packs: 40  
Revenue: 820  
---  
Enter choice (1-8): 8  
---  
No orders left to prepare.  
Exiting...
```

6. Cancel an order

```
Enter menu filename (e.g., menu.txt): MondayMenu.txt  
The drink Cappuccino is added to the menu.  
The drink Flat-White is added to the menu.  
The drink Mocha is added to the menu.  
The drink Chai-Latte is added to the menu.  
The drink Iced-Coffee is added to the menu.  
The drink Macchiato is added to the menu.  
The drink Turkish-Coffee is added to the menu.  
The drink Caramel-Macchiato is added to the menu.
```

Tab 1

Headings you add to the document will appear here.

```
The drink Matcha-Latte is added to the menu.  
The drink Hot-Chocolate is added to the menu.  
---  
Starbucks Order Management System  
---
```

Options:

1. Display today's Starbucks menu
2. Place a new order
3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)

```
Enter choice (1-8): 6
---
Enter Order No. to cancel: 100
No orders to cancel.
---
Enter choice (1-8): 6
---
Enter Order No. to cancel: -76667
No orders to cancel.
---
Enter choice (1-8): 2
---
Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Iced-Coffee 12000
Enter: enough
Order No. 100 added to the queue.
---
Enter choice (1-8): 2
---
Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Hot-Chocolate 10
Enter: Caramel-Macchiato 12
Enter: enough
Order No. 110 added to the queue.
---
Enter choice (1-8): 6
---
Enter Order No. to cancel: 130
Order No. 130 not found.
---
Enter choice (1-8): 6
---
Enter Order No. to cancel: 110
Order No. 110 canceled.
```

Tab 1

Headings you add to the document will appear here.

```
---
Enter choice (1-8): 4
---
Current orders:
Order No. 100
Drinks:
Iced-Coffee with 12000 sugar packs
Total Price: 145
---
Enter choice (1-8): 6
---
Enter Order No. to cancel: 100
Order No. 100 canceled.
---
Enter choice (1-8): 4
---
No orders in the queue.
---
Enter choice (1-8): 8
---
No orders left to prepare.
Exiting...
```

7. Rush an order

```
Enter menu filename (e.g., menu.txt): MondayMenu.txt
The drink Cappuccino is added to the menu.
The drink Flat-White is added to the menu.
The drink Mocha is added to the menu.
The drink Chai-Latte is added to the menu.
The drink Iced-Coffee is added to the menu.
The drink Macchiato is added to the menu.
The drink Turkish-Coffee is added to the menu.
The drink Caramel-Macchiato is added to the menu.
The drink Matcha-Latte is added to the menu.
The drink Hot-Chocolate is added to the menu.
---
Starbucks Order Management System
---
Options:
1. Display today's Starbucks menu
2. Place a new order
3. Prepare the next order
4. Display current orders in preparation queue
5. View Starbucks order report
6. Cancel an order
7. Rush an order
8. Exit (prepare all remaining orders first)
---
```

Tab 1

Headings you add to the document will appear here.

Tab 1

Headings you add to the document will appear here.

```
Enter choice (1-8): 7
---
Enter Order No. to rush: 150
No orders to rush.
---
Enter choice (1-8): 7
---
Enter Order No. to rush: 0
No orders to rush.
---
Enter choice (1-8): 2
---
Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Caramel-Macchiato 10
Enter: Mocha 60
Enter: Hot-Chocolate 0
Enter: Cappuccino
Invalid format. Enter name and sugar.
Enter: Cappuccino 1
Enter: enough
Order No. 100 added to the queue.
---
Enter choice (1-8): 2
---
Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Turkish-Coffee 1
Enter: Flat-White 0
Enter: turkish-coffee 0
Enter: enough
Order No. 110 added to the queue.
---
Enter choice (1-8): 2
---
Enter drink details (name sugar) one by one. Type 'enough' to stop.
Enter: Iced-Coffee 1
Enter: Iced-Coffee 2
Enter: enough
Order No. 120 added to the queue.
---
Enter choice (1-8): 4
---
Current orders:
Order No. 100
Drinks:
Cappuccino with 1 sugar packs
Hot-Chocolate with 0 sugar packs
Mocha with 60 sugar packs
Caramel-Macchiato with 10 sugar packs
Total Price: 670
```

Tab 1

Headings you add to the document will appear here.

```
---
Order No. 110
Drinks:
turkish-coffee with 0 sugar packs
Flat-White with 0 sugar packs
Turkish-Coffee with 1 sugar packs
Total Price: 415
---
Order No. 120
Drinks:
Iced-Coffee with 2 sugar packs
Iced-Coffee with 1 sugar packs
Total Price: 290
---
Enter choice (1-8): 7
---
Enter Order No. to rush: 130
Order No. 130 not found.
---
Enter choice (1-8): 7
---
Enter Order No. to rush: 110
Order No. 110 moved to front and rushed (price +30%).
---
Enter choice (1-8): 4
---
Current orders:
Order No. 110
Drinks:
turkish-coffee with 0 sugar packs
Flat-White with 0 sugar packs
Turkish-Coffee with 1 sugar packs
Total Price: 539.5
---
Order No. 100
Drinks:
Cappuccino with 1 sugar packs
Hot-Chocolate with 0 sugar packs
Mocha with 60 sugar packs
Caramel-Macchiato with 10 sugar packs
Total Price: 670
---
```

Tab 1

Headings you add to the document will appear here.

```
Order No. 120
Drinks:
Iced-Coffee with 2 sugar packs
Iced-Coffee with 1 sugar packs
Total Price: 290
---
Enter choice (1-8): 7
```

Tab 1

Headings you add to the document will appear here.

```
---
Enter Order No. to rush: 120
Order No. 120 moved to front and rushed (price +30%).
---
Enter choice (1-8): 4
---
Current orders:
Order No. 120
Drinks:
Iced-Coffee with 2 sugar packs
Iced-Coffee with 1 sugar packs
Total Price: 377
---
Order No. 110
Drinks:
turkish-coffee with 0 sugar packs
Flat-White with 0 sugar packs
Turkish-Coffee with 1 sugar packs
Total Price: 539.5
---
Order No. 100
Drinks:
Cappuccino with 1 sugar packs
Hot-Chocolate with 0 sugar packs
Mocha with 60 sugar packs
Caramel-Macchiato with 10 sugar packs
Total Price: 670
---
Enter choice (1-8): 7
---
Enter Order No. to rush: 120
Order No. 120 rushed (price +30%).
---
Enter choice (1-8): 4
```

```
---
Current orders:
Order No. 120
Drinks:
Iced-Coffee with 2 sugar packs
Iced-Coffee with 1 sugar packs
Total Price: 490.1
---
Order No. 110
Drinks:
```

```
turkish-coffee with 0 sugar packs
Flat-White with 0 sugar packs
Turkish-Coffee with 1 sugar packs
Total Price: 539.5
---
```

Tab 1

Headings you add to the document will appear here.

```
Order No. 100
Drinks:
Cappuccino with 1 sugar packs
Hot-Chocolate with 0 sugar packs
Mocha with 60 sugar packs
Caramel-Macchiato with 10 sugar packs
Total Price: 670
---
Enter choice (1-8): 3
---
```

```
Preparing Order No. 120
Total price: 490.1
Drinks (LIFO):
Iced-Coffee with 2 sugar packs
Iced-Coffee with 1 sugar packs
---
Enter choice (1-8): 8
---
Preparing remaining orders...
Preparing Order No. 110
Total price: 539.5
Drinks (LIFO):
turkish-coffee with 0 sugar packs
Flat-White with 0 sugar packs
Turkish-Coffee with 1 sugar packs
---
Preparing Order No. 100
Total price: 670
Drinks (LIFO):
Cappuccino with 1 sugar packs
```

Tab 1

Headings you add to the document will appear here.

```
Hot-Chocolate with 0 sugar packs
Mocha with 60 sugar packs
Caramel-Macchiato with 10 sugar packs
---
Exiting...
```

Tab 1

Headings you add to the document will appear here.

Some Important Rules

In order to get full credit, your program must be efficient, modular (with the use of functions), well commented and properly indented. Besides, you also have to use understandable identifier names. Presence of any redundant computation, bad indentation, meaningless identifiers or missing/irrelevant comments may decrease your grade in case that we detect them. When we grade your THEs, we pay attention to these issues. Moreover, **we may test your programs with very large test cases**. Hence, take into consideration the efficiency of your algorithms other than correctness.

Sample runs give a good estimate of how correct your implementation is, however, **we will test your programs with different test cases and your final grade may conflict with what you have seen on CodeRunner**. We will also **manually** check your code, indentations and so on, hence do not object to your grade based on the **CodeRunner** results, but rather, consider every detail on this documentation. **So please make sure that you have read this documentation carefully and covered all possible cases, even some other cases you may not have seen on CodeRunner or the sample runs**. The cases that you do not need to consider are also given throughout this documentation.

Submit via SUCourse ONLY! Paper, e-mail or any other methods are not acceptable.

The internal clock of SUCourse might be a couple of minutes skewed, so make sure you do not leave the submission to the last minute. In the case of failing to submit your THE on time:

"No successful submission on SUCourse on time = A grade of zero (0) directly."

What and where to submit (PLEASE READ, IMPORTANT)

It'd be a good idea to write your name and last name in the program (as a comment line of course). **Do not use any Turkish characters anywhere in your code (not even in comment parts)**. If your full name is "Duygu Karaoglan Altop", and if you want to write it as comment, then you must type it as follows:

// Duygu Karaoglan Altop

You should copy the full content of the .cpp file and paste it into the specified "Answer" area in the relevant assignment submission page on SUCourse. **Please note that the warnings are also considered as errors on CodeRunner, which means that you should have a compiling and warning-free program.**

Since the grading process will be automatic, you are expected to strictly follow these guidelines. **If you do not follow these guidelines, your grade will be zero (0)**. Any tiny change in the output format will result in your grade being zero (0), so please test your programs yourself, and against the sample runs that are available at the relevant assignment submission page on SUCourse.

Tab 1

Headings you add to the document will appear here.

In the CodeRunner, there are some visible and invisible (hidden) test cases. You will see your final grade (including hidden test cases) before submitting your code. There is no re-submission. You don't have to complete your task in one time, you can continue from where you left last time but you should not press submit before finalizing it. Therefore, you should make sure that it's your final solution version before you submit it. Also, we still do not suggest that you develop your solution on CodeRunner but rather on your IDE on your computer.

You may visit the office hours if you have any questions regarding submissions.

How to get help?

You may ask your questions to TAs or to the instructor. Information regarding the office hours of the TAs and the instructor are available at SUCourse.

Plagiarism

Tab 1

:

Headings you add to the document
will appear here.

Plagiarism is checked by automated tools, and we are very capable of detecting such cases. Be careful with that. Exchange of abstract ideas are totally okay but once you start sharing the code with each other, it is very probable to get caught by plagiarism. So, do NOT send any part of your code to your friends by any means or you might be charged as well, although you have done your THE by yourself. THEs are to be done personally and you have to submit your own work. **Cooperation will NOT be counted as an excuse.**

In case of plagiarism, the rules on the Syllabus apply.

Tab 1

:

Headings you add to the document
will appear here.

Good Luck!

Parسا