CSE108 – Computer Programming Lab.

Lab 14

Due at 10am.

Hand in: A student with number 20180000001 should hand in a zip file named 20180000001.zip for this lab.

Serialization LIFO and FIFO Data Structures from Binary File

In this assignment, you will recall the FIFO (First-In-First-Out) and LIFO (Last-In-First-Out) data structures you developed in Lab 11. This time, you need to create a function that restores the LIFO data structure from a binary file.

- **[10 point]** Implement the enqueue function, which adds a new order to the linked list. The function should take the following parameters : **void enqueue(Order* list, char* customerName, int orderID, char* items, time_t orderTime)**
  This function should add a new order in the linked list. The function should take the following parameters:
  Order* list: Pointer to the starting node representing the linked list.
  char* customerName: Customer name.
  int orderID: Order ID.
  char* items: Ordered items.
  time_t orderTime: Order time.

- **[10 point]** Implement the dequeue function, which removes and returns the oldest order from the linked list. However, add an additional parameter to the dequeue function to specify a time threshold. The function should remove and return all orders that are older than the specified time threshold. The function should take the following parameters: **Order* dequeue(Order* list, time_t thresholdTime)**
  This function should remove and return the oldest order from the linked list. However, add an additional parameter to the dequeue function to specify the time threshold. The function should remove and return all orders older than the specified time threshold. The function should take the following parameters:

  Order* list: Pointer to the starting node representing the linked list.
  time_t thresholdTime: The time threshold.

- **[25 point]** Implement the serializeLIFO function, which serializes the LIFO data structure to a binary file. The function should take the following parameters: **void serializeLIFO(LIFO* stack, const char* filename)**

This function should serialize the LIFO data structure into a binary file. The function should take the following parameters:

LIFO* stack: Pointer to the LIFO data structure to serialize.
const char* filename: The filename.

- **[15 point]** Implement the deserializeLIFO function, which deserializes the LIFO data structure from a binary file. The function should return a pointer to the deserialized LIFO structure. The function should take the following parameters: **LIFO* deserializeLIFO(const char* filename)**
  This function should decode the LIFO data structure back from a binary file. The function must return a pointer to the LIFO structure that is resolved back. The function should take the following parameters:
  const char* filename: The filename.
- **[25 point]** Implement the serializeFIFO function, which serializes the FIFO data structure to a binary file. The function should take the following parameters: **void serializeFIFO(FIFO* queue, const char* filename)**
  This function should serialize the FIFO data structure into a binary file. The function should take the following parameters:

  FIFO* queue: Pointer to the FIFO data structure to serialize.
  const char* filename: The filename.
- **[15 point]** Implement the deserializeFIFO function, which deserializes the FIFO data structure from a binary file. The function should return a pointer to the deserialized FIFO structure. The function should take the following parameters: **FIFO* deserializeFIFO(const char* filename)**

  This function should decode the FIFO data structure back from a binary file. The function must return a pointer to the FIFO structure that is resolved back. The function should take the following parameters:

  const char* filename: The filename.

Example Output:

```
enqueue(list, "Ali Yılmaz", 101, " Pizza, Salad, Ice Cream ", time(NULL));

Order Added: Customer Name: Ali Yılmaz, Order ID: 101, Items: Pizza, Salad, Ice Cream, Order Time: Sat Jun 01 09:30:00 2023

enqueue(list, "Ayşe Demir", 102, "Item2", time(NULL));

Order Added: Customer Name: Ayşe Demir, Order ID: 102, Items: Steak, Mashed Potatoes, Salad, Order Time: Sat Jun 01 09:35:00 2023
```

```
dequeue(list, thresholdTime);

Order Removed from Queue: Customer Name: Ali Yılmaz, Order ID: 101,
Items: Pizza, Salad, Ice Cream, Order Time: Sat Jun 01 09:30:00 2023


serializeLIFO(stack, "lifo_data.bin");

LIFO data structure serialized to a binary file.

deserializeLIFO("lifo_data.bin");

LIFO data structure deserialized from a binary file.


serializeFIFO(queue, "fifo_data.bin");

FIFO data structure serialized to a binary file.

deserializeFIFO("fifo_data.bin");

FIFO data structure deserialized from a binary file.
```

General Rules:

1. You will have two hours to provide a solution to the given problem set.
2. You will be able to hand in your solutions via Teams in the next two hours. The submission will be closed exactly at 10am.
3. There will be an interview session immediately after the submission deadline. Starting at 10am, you will be randomly invited to attend a meeting by a TA to demonstrate your solution and answer any questions asked by the TA.
4. You must be available until 1pm to respond to the demo invitation whenever you receive it. You will have 3 minutes after you are called via Teams. If you do not answer/appear in 3 minutes, you will miss you interview.
5. If you miss your interview or are unable to give satisfactory answers to the questions, you will receive a zero for that lab even if you have submitted your solution.
6. If you have not submitted a solution in time, you will not be invited for the interview and receive zero for that lab.
7. Due to time constraints, some students may not be invited to an interview. In that case, their solutions will be graded offline.
8. Unless you aren't declared for a specific prototype, you may use arbitrary but proper function and variable names that evoke its functionality.
9. The solution must be developed on given version of OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.

10. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it is working partially.
11. Zip your solution file before uploading it to MS Teams. The zip file must contain the C file with your solution and screenshots of the valid outputs of the program.