

# Wenyue Wang – HW2 CSE 6010

- For this homework, I started by creating four files, a .h file to include the definition of LLQueue struct as well as include all functions, an assignment2.c file to include all functions for the LLQ like create, insert, delete, search, etc, a unittest.c file to define all test cases into multiple functions and a main.c file to call these tests. Then I defined the LLQueue struct which include the value of current node and address of next node. Then I fill in required operations of LLQ and generated multiple tests including edge cases. Then I call all test cases in main.c file to generate the final result.

# Test cases (5 in total)

- Create and insert test: in this test, I first call create function to create a blank linked list, then insert one item to the blank list. After that I insert another 4 elements to ensure that there are 5 elements in the queue in total, so that I can insert another number. In addition, I called LLQ\_count and LLQ\_print to count element numbers and print out results.
- Max and min test: in this test, I generated the same linked list as before, and perform LLQ\_maximum and LLQ\_minimum function to return the max and min element.
- Search test: in this test, I still generated the same linked list as before, then perform LLQ\_search to search for item that appears at the beginning, middle and end of the queue. In addition, I also tested to search a number that is not in the queue.
- Delete test: in this test, based on the generated queue, I popped the element one by one from the beginning of the linked list, and I print out which element I popped out and what the linked list looked like after deleting. In addition, I also tested what will happen if I try to delete element from an empty linked list, and it shown as error and returned value is 0.0.
- Free test: in this test, I tested whether the linked-list has been totally deleted and free up memories. Here I tested two conditions, the input linked list with elements and an empty linked list. And after the test, both cases have been verified as deleted and free up space, (cannot print that linked list after calling free function because the linked does not exist any more).

# Screenshot of test results

```
----- CREATE AND INSERT TEST -----
Test create a new queue:
Successfully create a new queue!

Test insert an item into an empty queue:
Before inserting, the queue is: The linked list queue is empty
And number of elements in this case is: 0.
After inserting, the queue is: 50.00
And number of elements in this case is: 1.

Test insert an item into a queue containing at least 4 elements:
Before inserting, the queue is: 50.00 34.24 66.00 22.60 99.99
And number of elements in this case is: 5.
After inserting, the queue is: 50.00 34.24 66.00 22.60 99.99 63.00
And number of elements in this case is: 6.

----- MAX AND MIN TEST -----
Successfully create a new queue!
The original queue is:
50.00 34.24 66.00 22.60 99.99 63.00
min value is: 22.60
max value is: 99.99

----- SEARCH TEST -----
Successfully create a new queue!
50.00 34.24 66.00 22.60 99.99 63.00
search result for front element is: 0x7fe46c6058d0
search result for middle element is: 0x7fe46c6058f0
search result for end element is: 0x7fe46c605920
search result for not present element is: 0x0
```

```
----- DELETE TEST -----
Successfully create a new queue!
Deleting all items in a queue one after the other:
The original queue is:
50.00 34.24 66.00 22.60 99.99 63.00
1's deleted node is: 50.00
linked list now is: 34.24 66.00 22.60 99.99 63.00
2's deleted node is: 34.24
linked list now is: 66.00 22.60 99.99 63.00
3's deleted node is: 66.00
linked list now is: 22.60 99.99 63.00
4's deleted node is: 22.60
linked list now is: 99.99 63.00
5's deleted node is: 99.99
linked list now is: The linked list queue is empty63.00
6's deleted node is: 63.00
linked list now is: The linked list queue is empty

Now if the queue is empty:
Successfully create a new queue!
The linked list queue is empty
Error! Invalid linked list queue.
deleted node is: 0.00
linked list now is: The linked list queue is empty

----- FREE TEST -----
Successfully create a new queue!
Before free up the queue, the LLQ is: 50.00 34.24 66.00 22.60 99.99 63.00
Has free up all memory space

Now if the input queue is empty:
Successfully create a new queue!
Before free up the queue, the LLQ is: The linked list queue is empty
Has free up all memory space

[Process completed]
```

# Most challenge part:

- The most challenge part for this hw is how to free up the space of my linked-list, and how to test if my free function is correct or not. It's my first time learn how to free up the space, so I searched online about how to give the location and how to free up the location, and the most common mistake I meet is that sometimes I freed up the space before I actually use it, especially when it's in a while loop, like free up each node one by one.