

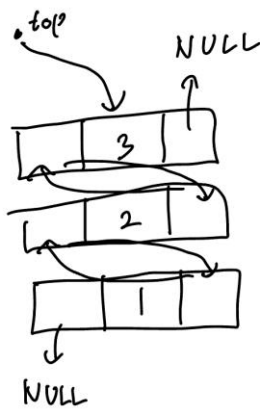
# Assignment 3 Technical Report

2171039 Chaewon Lee

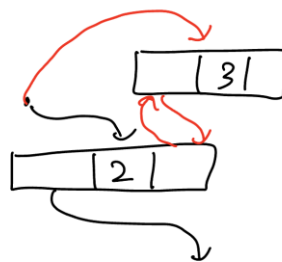
## # Code 1

This code is an implementation of stack with circular doubly linked list.

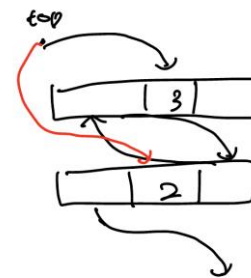
To draw the stack that is implemented in the main function, it would look like the picture 1.



[Picture 1]



[Picture 2]



[Picture 3]

In push function, first it creates a temp node. And, it set the item inside the temp node same as the item that push function received by its argument. Next, the function calls is\_empty function to check whether the stack is empty or not. If the stack is empty, every link of the temp should be null. And if it is not, the formal top node's right link would be new top node, and new top node's left link would be formal top node. Lastly, top node should point new top node. You can see this situation as an illustration in picture 2. A dot means top pointer, 2 is former top node, and 3 is new top node. Black links are formal links, and red links are new links.

In pop function, first it sets temp node as formal top node. Then it checks the temp node is the only node in the stack or not by checking its left link. If temp's left link is NULL, it means that it is the only node in the stack. When there are more nodes in the stack, then the code sets new top node's right link to null. Then, commonly, the code set top of the stack as the new one, and free the temp node. You can see how this function works when there are more than two elements in the stack with picture 3.

## - Result

```
3
2
1
Process finished with exit code 0
```

## # Code 2

This code is a modified version of Simulation.cpp with two bank staffs.

To increase the number of staffs from one to two, there are two service time variable. The service time which staff 1 has is `service_time_1`, and staff 2 has is `service_time_2`. And, while the clock is smaller than the max duration, the code checks whether each staffs has their customer or not. If they do not have any customer, customer is assigned with `remove_customer` function. And after the assignment, if the service time is bigger than 0, a message that the staff accepted the customer is printed to make it clear that which staff accepted the customer. But this message has a flaw, since the service time assigned is same as 0 when there is no customer left in the queue and the original service time was 1. (`remove_customer` function decreases the service time and assigns, so the time assigned would be 0.) To fix this, there should be another variable that saves whether the queue is empty or not, but since this does not affect the total result of the program, I kept it in this way.

Other parts of the code, such as queue and the functions about it, `insert_customer`, `remove_customer`, `print_stat` function is same as the original code.