

Name: 周嘉禾

Student ID: D1166506

In this program, I separate the requirement into 3 parts, which are main function, IQueue class, and Node class, respectively.

### **Node class**

This class is the most fundamental part to construct the double-linked linear list. Inside this class, we have:

- Friend class IQueue, which can access the private data of the Node class.
- Private:
  1. elem: record the data of a node in integer type.
  2. prev: a pointer pointing to the previous node.
  3. next: a pointer pointing to the next node.
- Public:
  1. Node(): the default constructor which will set both prev and next pointer as NULL and elem as 0.
  2. Node(int): the constructor with data and will set both prev and next pointer as NULL and elem as the number given.

### **IQueue class**

This class provides the necessary operations we need to run double-linked linear lists. Inside this class, we have:

- Private:
  1. head: a pointer pointing to the first node in the double-linked linear list.
  2. tail: a pointer pointing to the last node in the double-linked linear list.
- Public:
  1. IQueue(): default constructor which will set both head and tail pointer as NULL.
  2. void push(int): push the given element into the last of the list and update the pointer of the node and the tail.
  3. int pop(): pop the element out from head.
  4. int front(): check front element of the queue.
  5. int back(): check back element of the queue.

6. `bool isEmpty()`: check whether the queue is empty.
7. `int getSize()`: get the size of the queue.
8. `Node *getHead()`: get the head pointer of the queue, i.e., the front pointer.
9. `Node *getTail()`: get the tail pointer of the queue, i.e., the back pointer.
10. `void printHeadToTail()`: print the queue from head to tail.

### **Main function**

This part is much easier than previous since we just have to implement the main function as the requirement. Therefore, I use random generator to generate the trail count, enqueue count, and dequeue count, declare the double-linked linear list before we go further, and call the function needed to complete the program, like calling the enqueue, dequeue, and `printHeadToTail`.