HOW TO USE QUEUE IN STANDARD TEMPLATE LIBRARY

Queues are a type of **container adaptor**, which is implemented to operate in a FIFO (First in, First out). Elements are pushed into the "back" of the specific container and popped from its "front".

Declare: `#include <queue>`

```
#include <queue>
Template: stack<value_type> queue_name;
```

Member function:

- size: return the current size of the queue. Complexity: O(1).
- empty: return true if queue is empty; otherwise, return false. Complexity: O(1).
- push: insert an element at the end the queue. Complexity: O(1).
- pop: remove the element on the top of the queue. Complexity: O(1).
- front: return the value of element at the top of the queue. Complexity: O(1).
- back: return the value of element at the back of the queue. Complexity: O(1).

Demo program:

```
#include <iostream>
using namespace std;

#include <queue>

int main() {
	queue<int> q;
	for (int i = 1; i <= 5; ++i) q.push(i); // q = {1, 2, 3, 4, 5}
	q.push(10); // s = {1, 2, 3, 4, 5, 10}
	cout << q.size() << endl; // Print out on the screen: 6
	cout << q.front() << endl; // Print out on the screen: 1
	q.pop(); // s = {2, 3, 4, 5, 10}
	cout << q.back() << endl; // Print out on the screen: 10
	cout << q.empty() << endl; // Print out: 0 (since queue s is not empty).
	return 0;
}
```