Queue

Queues are a type of container adapters that operate in a **first in first out (FIFO)** type of arrangement. Elements are inserted at the back (end) and are deleted from the front. Queues use an encapsulated object of deque or list (sequential container class) as its underlying container, providing a specific set of member functions to access its elements.

1. To use a queue, you have to include the 'queue' header file

```
#include <queue>
```

2. The syntax to define a queue is:

```
std::queue<data_type> queueName;
```

Functions

The functions are very similar to stack. Some of them are:

- **empty()** - Returns whether the queue is empty. It return true if the queue is empty otherwise returns false.|
- **size()** - Returns the size of the queue.|
- **swap(g)** - Exchange the contents of a queue with another queue 'g' but the queues must be of the same data type, although sizes may differ.|
- **emplace()** - Insert a new element into the queue container, the new element is added to the end of the queue.|
- **front()** - Returns a reference to the first element of the queue.|
- **back()** - Returns a reference to the last element of the queue.|
- **push()** - Adds the element ‘g’ at the end of the queue.|
- **pop()** - Deletes the first element of the queue.|

All these operations have O(1) time complexity.