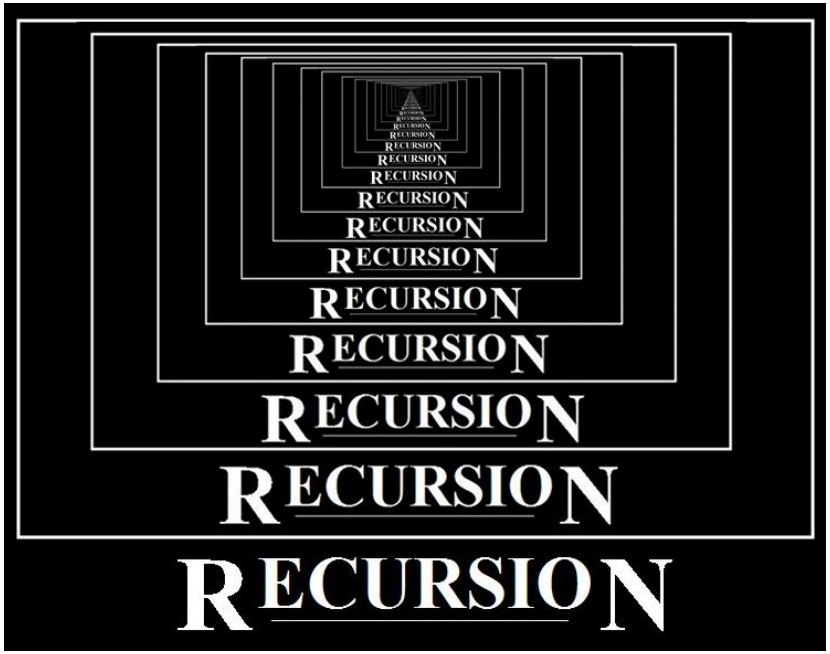
Welcome to CS 225

lab_quacks

lab_qu(eue+st)acks

Spiteful Stacks and Questionable Queues

- Recursion
- Stack
- Queue



The process in which a function calls itself directly or indirectly is called **recursion** and the corresponding function is called a **recursive function**.

A common computer programming tactic is to divide a problem into sub-problems of the same type as the original, solve those sub-problems, and combine the results.

https://en.wikipedia.org/wiki/Recursion_(computer_science)

Recursive Functions

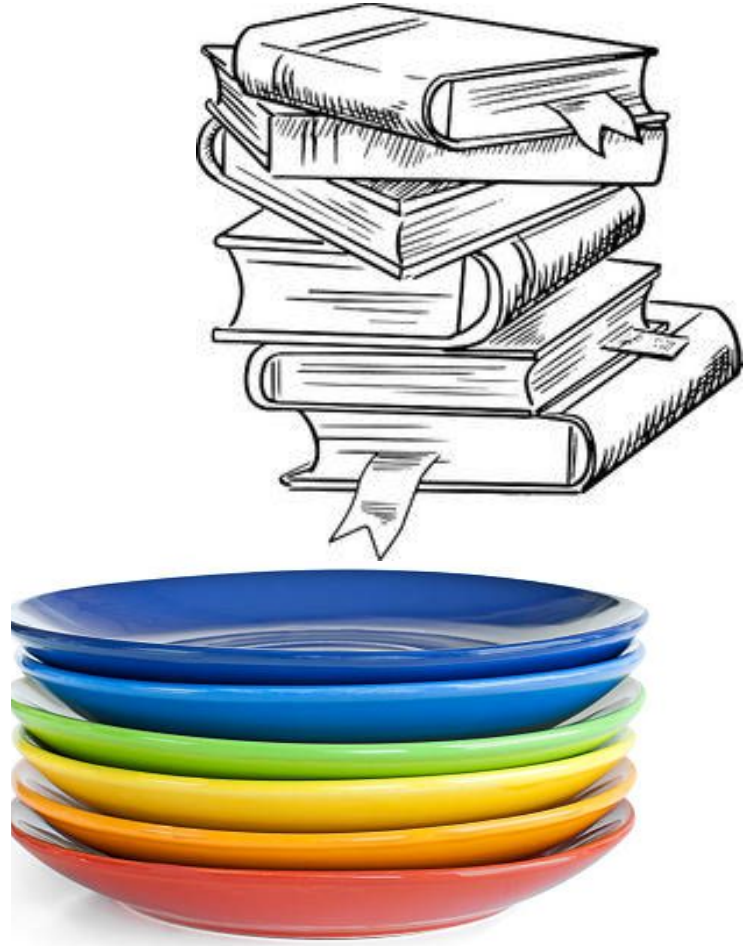
1. Base Case
2. Recursive Step
3. Incremental Step

```
1  int factorial(int n)
2  {
3      int result = 1;
4      for (int i = 1; i <= n; i++)
5          result = result * i;
6      return result;
7  }
8
9  int factorial(int n)
10 {
11    if(n == 0) return 1;
12    return n*factorial(n-1);
13
14 }
15
16
17
18
19
20
```

```
1  int factorial(int n)
2  {
3      if (n == 0) return 1;
4      return factorial(n-1) * n;
5  }
6
7
8
```

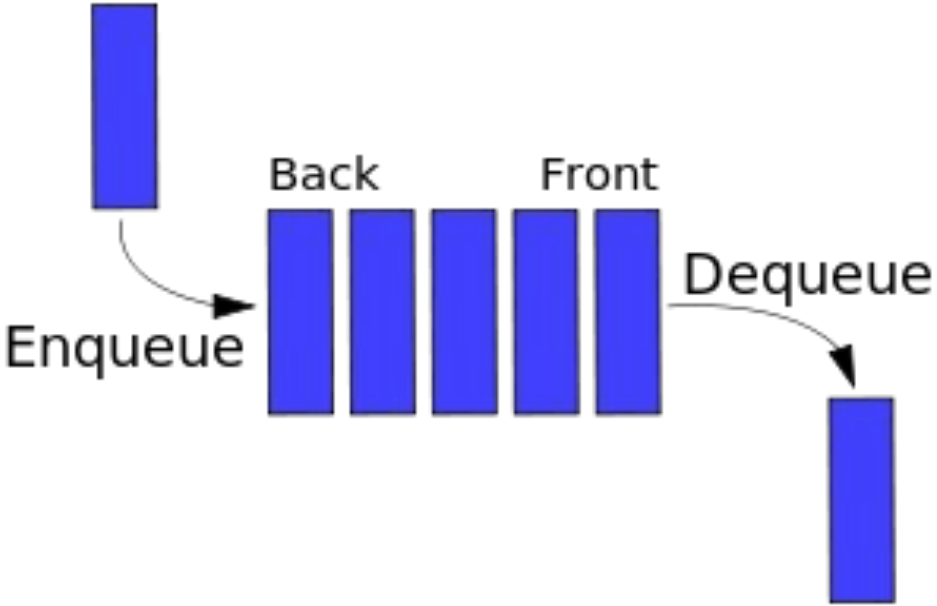
$$n! = \begin{cases} 1 & if\ n = 0 \\ n * (n-1)! & if\ n > 0 \end{cases}.$$

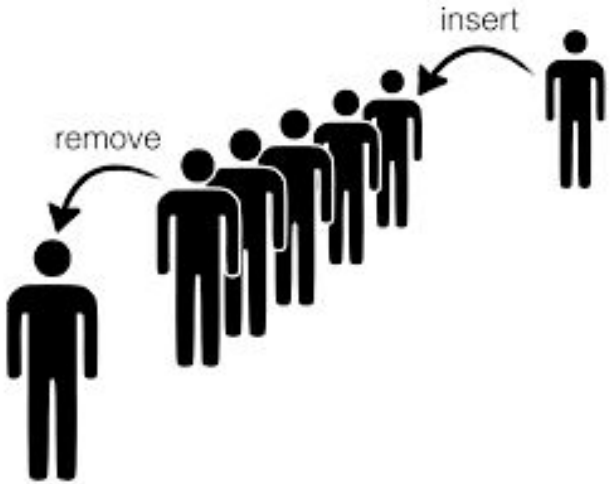
Queue and Stack



Queue

A Queue is a linear structure which follows a particular order in which the operations are performed.
The order is **First In First Out (FIFO).**





class template

std::queue

<queue>

```
template <class T, class Container = deque<T> > class queue;
```

FIFO queue

queues are a type of container adaptor, specifically designed to operate in a FIFO context (first-in first-out), where elements are inserted into one end of the container and extracted from the other.

fx Member functions

empty	Test whether container is empty (public member function)
size	Return size (public member function)
front	Access next element (public member function)
push	Insert element (public member function)
pop	Remove next element (public member function)

http://www.cplusplus.com/reference/queue/queue/

std::queue::pop

<queue>

```
void pop();
```

Remove next element

Removes the next element in the queue, effectively reducing its size by one.

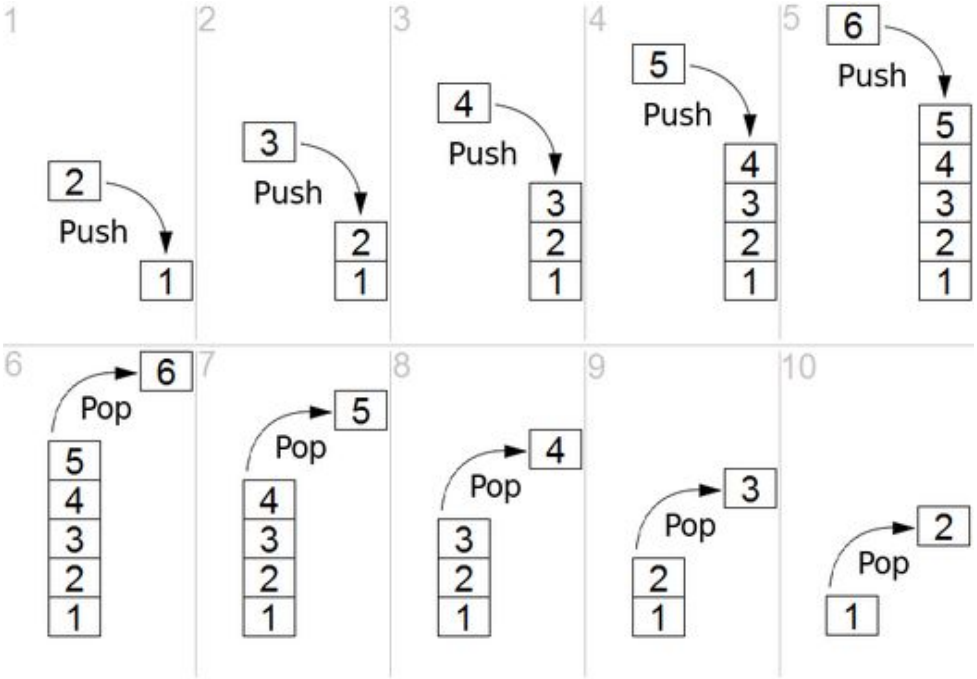
The element removed is the "oldest" element in the queue whose value can be retrieved by calling member queue::front.

Note that the `pop()` operations do not return the element removed!

Stack

Stack is a linear data structure which follows a particular order in which the operations are performed.

The order is LIFO(Last In First Out)



class template

std::stack

<stack>

```
template <class T, class Container = deque<T> > class stack;
```

LIFO stack

Stacks are a type of container adaptor, specifically designed to operate in a LIFO context (last-in first-out), where elements are inserted and extracted only from one end of the container.

fx Member functions

empty	Test whether container is empty (public member function)
size	Return size (public member function)
top	Access next element (public member function)
push	Insert element (public member function)
pop	Remove top element (public member function)

public member function

std::stack::pop

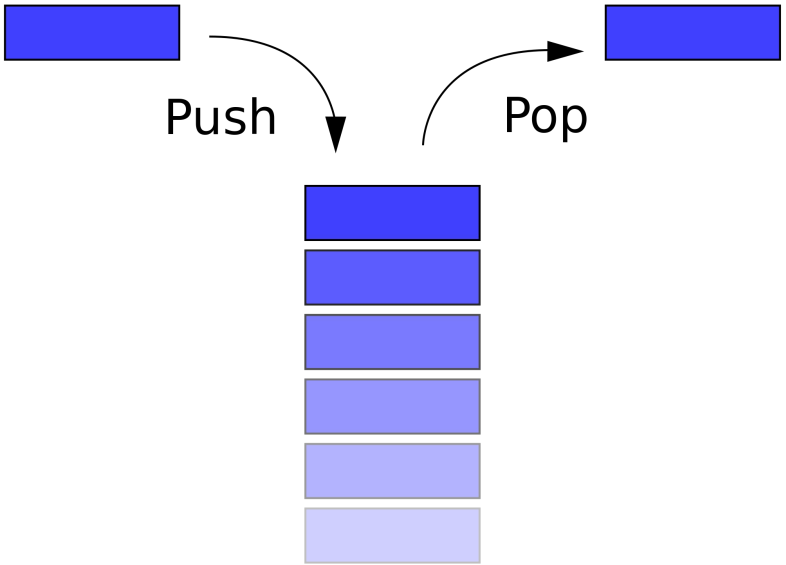
<stack>

```
void pop();
```

Remove top element

Removes the element on top of the stack, effectively reducing its size by one.

The element removed is the latest element inserted into the stack, whose value can be retrieved by calling member stack::top.



Common methods of enumerating elements in a Queue/Stack

```
1  void popAll(std::queue<int>& q){
2   while (!q.empty()) {
3    std::cout << q.front() << " ";
4     q.pop();
5   }
6   std::cout << std::endl;
7  }
8
```

```
1  void popAll(std::stack<int>& s){
2   while (!s.empty()) {
3    std::cout << s.top() << " ";
4     s.pop();
5   }
6   std::cout << std::endl;
7  }
8
9
```