List Implementation #2: ______________

```cpp
#pragma once
template <typename T>
class List {
public:
/* ... */
private:
};
```

Implementation Details and Analysis:
What is the running time of `insertFront()`?

> What is our resize strategy?

Array Resize Strategy #1:

...total copies across all resizes: _________

...total number of insert operations: _________

...average (amortized) cost of copies per insert: _________

Running Time:

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Singly Linked List</th>
<th>Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert/Remove at front</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert after a <code>given</code> element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove after a <code>given</code> element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert at <code>arbitrary</code> location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove at <code>arbitrary</code> location</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stack ADT

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Queue ADT

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stack and Queue Implementations

```
#pragma once
#include <vector>
template <typename T>
class Stack {
public:
    void push(const T & d);
    T pop();
    bool isEmpty();
private:
    std::vector<T> list_;}
#include "Stack.hpp"
```

```
Example 1
Queue<int> q;
q.enqueue(3);
q.enqueue(8);
q.enqueue(4);
q.dequeue();
q.enqueue(7);
q.dequeue();
q.dequeue();
q.enqueue(2);
q.enqueue(1);
q.enqueue(3);
q.enqueue(5);
q.dequeue();
q.enqueue(9);
```

```
Example 2
Queue<char> q;
q.enqueue('m');
q.enqueue('o');
q.enqueue('n');
...
q.enqueue('d');
q.enqueue('a');
q.enqueue('h');
q.enqueue('a');
```

Accessing Every Element in Our List / Queue / [Anything]
Suppose we want to look through every element in our data structure. What if we don’t know what our data structure even looks like?

```
```

<table>
<thead>
<tr>
<th>CS 225 – Things To Be Doing:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.  mp_stickers due today</td>
</tr>
<tr>
<td>2.  Daily POTDs</td>
</tr>
</tbody>
</table>