A Linked List implementation of a List:

```
#pragma once

template <typename T>
class List {
public:
    /* ... */
private:
    class ListNode {
        public:
            const T & data;
            ListNode * next;
        ListNode(T & data) :
            data(data), next(nullptr) { }
    };
    ListNode *head_;  
    /* ... */
};
```

Implementing a basic List operation:

```
#include "List.h"

template <typename T>
T & List<T>::operator[](unsigned index) {
    
    } 
```

Finding in a list:

```
template <typename T>
typename List<T>::ListNode * & List<T>::_index(unsigned index) {
    
    } 
```

What is the return type of _index?

Building functionality with _index():

```
template <typename T>
T & List<T>::insert( const T & t, unsigned index) {
    
    } 
```

```
template <typename T>
T & List<T>::insert( const T & t, unsigned index) {
    
    } 
```
List.hpp

```cpp
template <typename T>
T & List<T>::remove(unsigned index) {
}
```

List Implementation #2: ______________

**Alternate List.h**

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

private:
... */
};
```

Array - Implementation Details:

1. What is the running time of `insertFront()`?

```
C  S  2  2  5
[0] [1] [2] [3] [4]
```

2. What is our resize strategy?

Resize Strategy #1:

```

```

Resize Strategy #2:

```

```

3. What is the running time of `get()`?

```
```

CS 225 – Things To Be Doing:

1. Programming Exam A starts Sept. 27 (8 days from today)
2. MP2 due Sept. 24 (5 days from now); EC worth +5 tonight!
3. Lab Extra Credit → Attendance in your registered lab section!
4. Daily POTDs

```