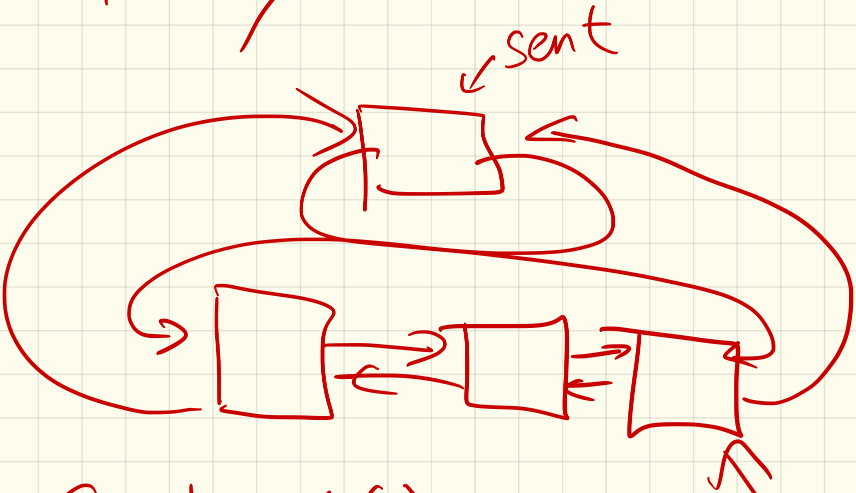
Recap

- HW 5 over vectors, will be up by tonight
- Reeding over rest of lists for tomorrow
  ↳ by 2pm
- No class next Friday (just before break)

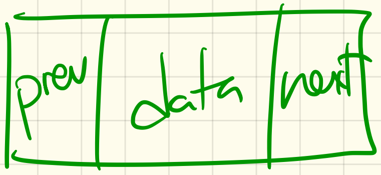
Today: back to Lists!

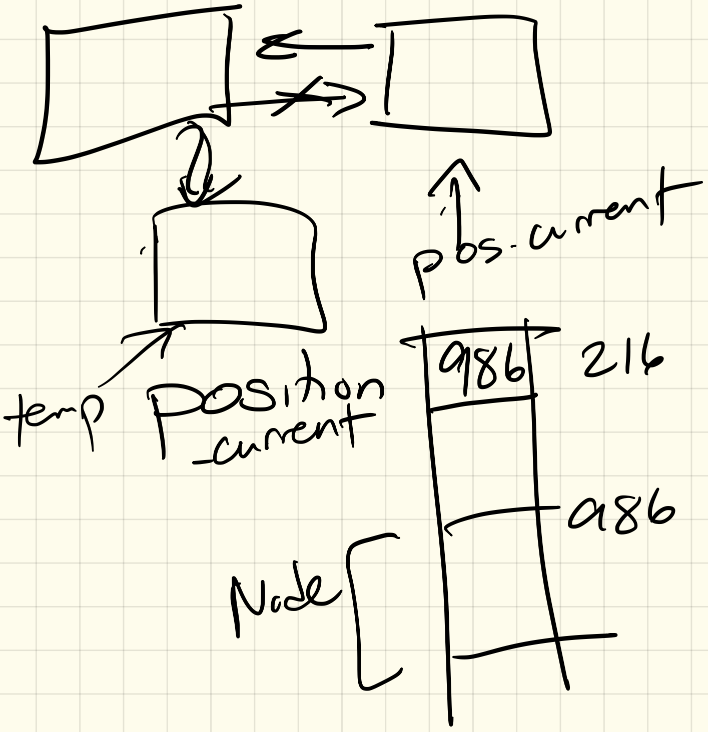
Doubly Linked lists:



Goal: O(1) insertion

insert before it





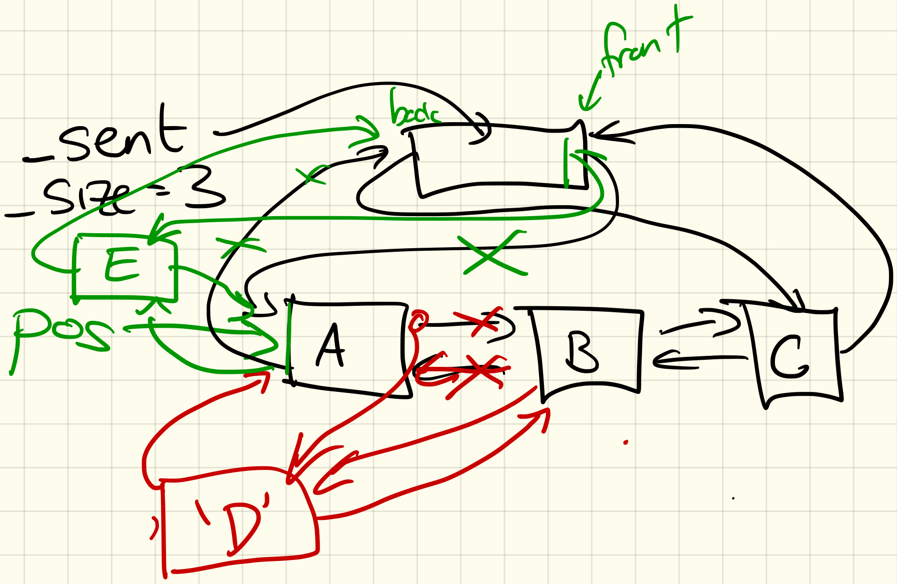
iterators are friends with lists:

```
(position._current)->
    _prev->_next = temp;
```

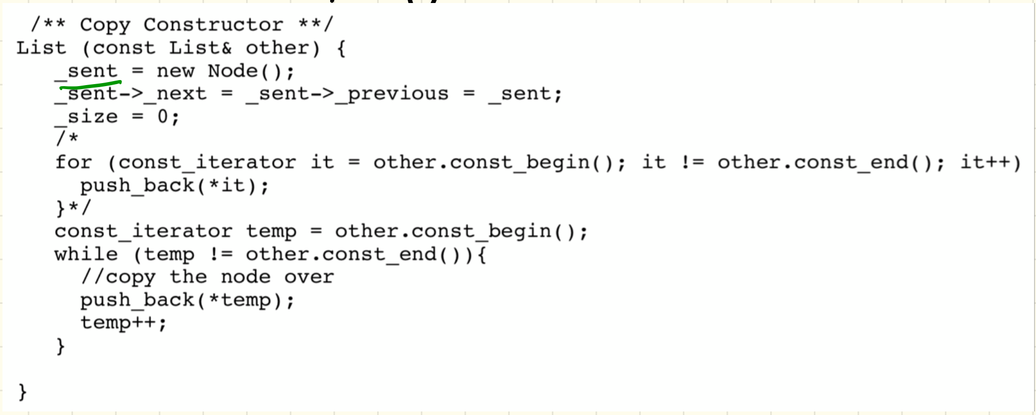
Solution:

```
/**
 * Function to insert into the list at a given spot
 *
 * Parameter position: an iterator we wish to insert before
 * Parameter element: the value to insert
 */
void insert(iterator position, T element) {
  Node* pos = position._current; ← optional
  Node* newnode = new Node(element, pos->_previous, pos);
  pos->_previous->_next = newnode;
  pos->_previous = newnode;
  _size++;
}
```

Picture:



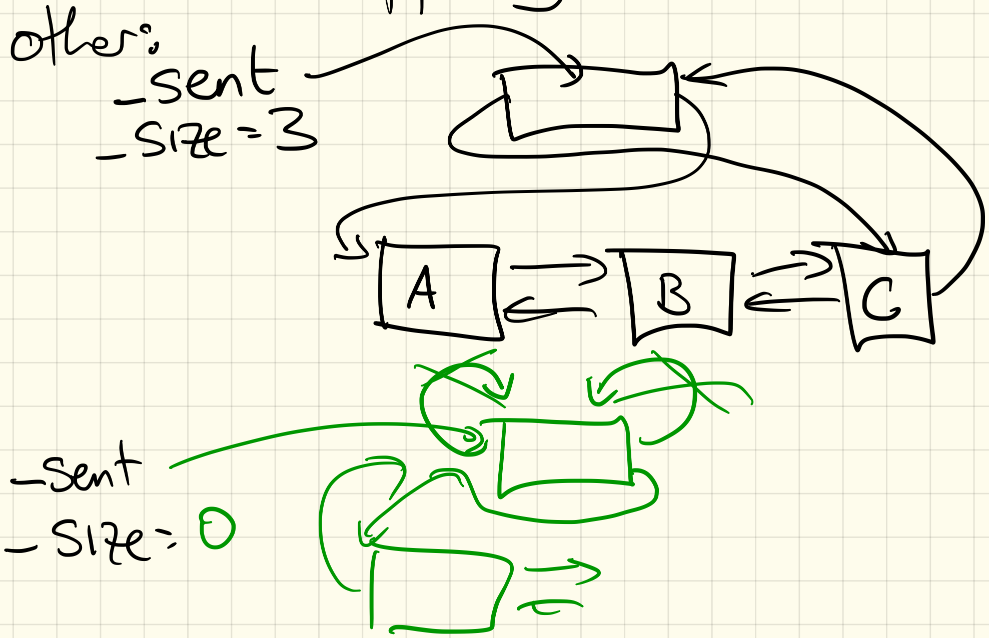
Housekeeping:



```
 /** Copy Constructor **/
List (const List& other) {
   _sent = new Node();
   _sent->_next = _sent->_previous = _sent;
   _size = 0;
   /*
   for (const_iterator it = other.const_begin(); it != other.const_end(); it++)
     push_back(*it);
   }*/
   const_iterator temp = other.const_begin();
   while (temp != other.const_end()){
     //copy the node over
     push_back(*temp);
     temp++;
   }

}
```

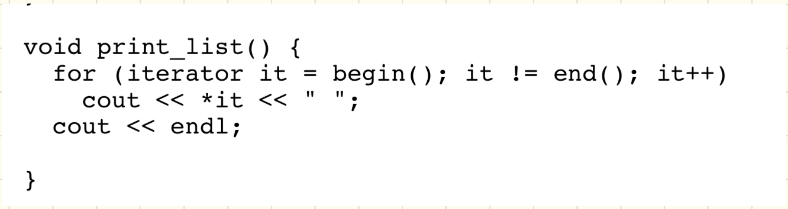
What is happening???



Other things:

Print:

Useful to dump entire list:



```
void print_list() {
  for (iterator it = begin(); it != end(); it++)
    cout << *it << " ";
  cout << endl;

}
```

testList.cpp: