

## Drill

1. Define an array of `ints` with the ten elements { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }.
2. Define a `vector<int>` with those ten elements.
3. Define a `list<int>` with those ten elements.
4. Define a second array, vector, and list, each initialized as a copy of the first array, vector, and list, respectively.
5. Increase the value of each element in the array by 2; increase the value of each element in the vector by 3; increase the value of each element in the list by 5.
6. Write a simple `copy()` operation.

```
template<typename Iter1, typename Iter2>  
    // requires Input_iterator<Iter1> && Output_iterator<Iter2>()  
Iter2 copy(Iter1 f1, Iter1 e1, Iter2 f2);
```

that copies `[f1,e1)` to `[f2,f2+(e1-f1))` and returns `f2+(e1-f1)` just like the standard library `copy` function. Note that if `f1==e1` the sequence is empty, so that there is nothing to copy.

7. Use your `copy()` to copy the array into the vector and to copy the list into the array.
8. Use the standard library `find()` to see if the vector contains the value 3 and print out its position if it does; use `find()` to see if the list contains the value 27 and print out its position if it does. The “position” of the first element is 0, the position of the second element is 1, etc. Note that if `find()` returns the end of the sequence, the value wasn't found.

7. Find the lexicographical last string in an unsorted `vector<string>`.