Write a fake vector class. The class should use the built-in vector/list/array in your language of choice to effectively "wrap" inside of it an existing vector, while presenting a limited vector-like functionality to the user. Here is a TypeScript interface that represents what your class should look like:

interface Vector<T> extends Iterable<T> {  
 get(index: number);  
 set(index: number, value: T);  
 length: number;  
 push(value: T);  
 pop(): T;  
 insert(index: number, value: T);  
 // remember to implement the iterable functionality  
}

If you are working in another language, you may translate this interface into a C# interface, Java interface, VB interface, or C++ pure virtual class, because the class you write must implement/inherit the interface/class that is shown above. If you are programming in raw JavaScript (ES6/ES2015), you cannot implement the interface, but you should carefully program your class to work identically to the presented interface.

Please ensure you implement an iterator (or Enumerable, for C# programmers). You may use generators if your language supports them, or you may implement the iterator classes manually if you choose to do so.

-5 points, code does not provide a valid output if pushed or popped or inserted additional values.

-5 points, array initializes with hard-coded values.

-5 points, interface not implemented.

//////////////////////////////////////////////

In standard bubble sort, the algorithm makes multiple passes through the list, swapping items so that the smaller value comes first and the larger value comes second.

Implement a "bouncing" bubble sort. In this version of bubble sort, instead of making passes through a list that starts at the beginning and runs through to the end, you should reverse the direction of each pass. That is, if the first pass starts at the beginning of the list and runs through to the end, the second pass would run from the end of the list back to the beginning, and then the third pass would start at the beginning again.

Assume items in the list are of the type integer.

-5 points, your sorting algorithm doesn't follow basic bubble sort logic. It traverses and checks in the only unsorted parts of the list not through the whole list again and again.