Eva Lu is talking about how when insert and delete return the queue, it is a pair with a front and rear pointer where the front pointer points to the first pair in the list of the entire sequence of items in the queue. A list object always points to a single pair, where the `car` accesses the first item in the list, and the `cdr` accesses the rest of the list (another pair). Therefore, the front pointer, by pointing to the first pair in the list, contains and will display the entire sequence of items in the queue.

The last item isn't inserted twice, but rather it gets inserted into the list (as seen in the list that is in the `car` position) as a consequence of becoming the rear pointer (as seen in the `cdr` position of the returned queue). This is because `insert-queue!` sets the `cdr` of the previous rear pointer to be the new rear pointer. This appends the new rear pointer to the queue list, so it also appears in the front pointer list.

When the last item is deleted, the queue is now empty (seen in the `car` position which is now the empty list), but the item remains the rear pointer since `delete-queue!` doesn't change the rear pointer. This doesn't matter since `empty-queue?` only checks if the front pointer is empty.

To print the queue, just return the front pointer, since it points to the list containing all items in the queue.