1. i. Because min must return an int, q must contain at least one Integer to return
   ii. If min did not have to be in entries(q), then the method could just return the minimum value of Integer and still satisfy the ensures clause.

```
/**
 * Reports an array of two {@code int}s with the smallest and
the
 * largest integer in the given {@code Queue<Integer>}.
 *
 * @param q
 *            the queue of integer
 * @return an array of two {@code int}s with the smallest and
the
 *        largest integer in the given queue
 * @requires q /= empty_string
 * @ensures <pre>
 * { minAndMax[0], minAndMax[1] } is subset of entries(q) and
 * for all x: integer
 *   where (x in in entries(q))
 *    (minAndMax[0] <= x <= minAndMax[1])
 * </pre>
 */
private static int[] minAndMax(Queue<Integer> q) {

     int min = q.dequeue();
     q.enqueue(min);
     int max = min;

     for (int i = 0; i < q.length() - 1; i++) {
          int current = q.dequeue();
          q.enqueue(current);
          if (max < current) {
               max = current;
          }
          if (current < min) {
               min = current;
          }
     }
     int[] minMax = {min, max};
     return minMax;
}
```

```
/**
 * Reports an array of two {@code int}s with the smallest and
the
 * largest integer in the given {@code Queue<Integer>}.
 *
 * @param q
 *            the queue of integer
 * @return an array of two {@code int}s with the smallest and
the
 *        largest integer in the given queue
 * @requires q /= empty_string
 * @ensures <pre>
 * { minAndMax[0], minAndMax[1] } is subset of entries(q) and
 * for all x: integer
 *   where (x in in entries(q))
 *   (minAndMax[0] <= x <= minAndMax[1])
 * </pre>
 */
private static int[] minAndMax(Queue<Integer> q) {

     int min = q.dequeue();
     q.enqueue(min);
     int max = min;

     for (int i = 0; i < q.length() - 1; i = i + 2) {
          int first = q.dequeue();
          int second = q.dequeue();
          q.enqueue(first);
          q.enqueue(second);
          if (first < second) {
               if (first < min) {
                    min = first;
               }
               if (second > max) {
                    max = second;
               }
          } else {
               if (second < min) {
                    min = first;
               }
               if (first > max) {
                    max = second;
               }
          }
     }
     int[] minMax = {min, max};
     return minMax;
}
```