Assignment 1

Consider an array \( b \) with \( n \) integer elements. Let \( j \) and \( k \) be two indices with \( 0 \leq j < k < n \). In the questions below, \( b[j..k] \) is a shortcut for a segment of the array \( b \), but in JML it cannot be used outside of assignable clauses. Write JML expressions which precisely describe the following:

1. All elements in \( b[j..k] \) are zero.
2. All zeros of \( b[0..n-1] \) are in \( b[j..k] \).
3. It is not the case that all zeros of \( b[0..n-1] \) are in \( b[j..k] \). Write this in two ways, once using negation ('!') and once avoiding negation.
4. \( b[0..n-1] \) contains two zeros.
5. \( b[0..n-1] \) contains at least two zeros.
6. \( b[0..n-1] \) contains at most two zeros.
7. Specify a method public static void reverse(int[] b) which reverses the order of elements in \( b \).
8. Specify a method public static int sum(int[] b) which computes the sum of all positive elements in \( b \).

Assignment 2

The following class implements selection sort. Specify the class in JML as completely as possible. Verify the code of the method min_index() against your specification with KeY.

```java
public class Sort {

    public static void selectionSort(int[] a) {
        for (int i = 0; i < a.length; i++) {
            int min_i = min_index(a, i);

            int temp = a[i];
            a[i] = a[min_i];
            a[min_i] = temp;
        }
    }

    private static int min_index(int[] a, int start_i) {
        int min_i = start_i;

        for (int i = start_i; i < a.length; i++) {
            if (a[i] < a[min_i]) min_i = i;
        }
        return min_i;
    }
}
```
return min_i;
}
}