1.

```
    /**
     * Finds {@code x} in {@code q} and, if such exists, moves it to the
front
     * of {@code q}.
     *
     * @param <T>
     *            type of {@code Queue} entries
     * @param q
     *            the {@code Queue} to be searched
     * @param x
     *            the entry to be searched for
     * @updates q
     * @ensures <pre>
     * perms(q, #q)  and
     * if <x> is substring of q
     *  then <x> is prefix of q
     * </pre>
     */
    private static <T> void moveToFront(Queue<T> q, T x) {
        assert q != null : "Violation of: q is not null";

        T temp = null;
        T item = null;
        Queue<T> tempQ = q.newInstance();

        for (int i = 0; i < q.length(); i++) {
            item = q.dequeue();
            if (x.equals(item)) {
                q.clear();
                q.enqueue(item);
                for (int j = 0; j < q.length(); j++) {
                    temp = tempQ.dequeue();
                    if (temp != item) {
                        q.enqueue(temp);
                    }
                }
            } else {
                q.enqueue(temp);
            }

        }

    }
```

2.

```
    @Test
    private void addTest() {
        Set<String> test = this.createFromArgsTest("1", "2", "3", "4");
        Set<String> ref = this.createFromArgsRef("1", "2", "3", "4", "5");

        test.add("5");
```

```
        assertEquals(ref, test);
    }

    @Test
    private void removeTest() {
        Set<String> test = this.createFromArgsTest("1", "2", "3", "4", "5");
        Set<String> ref = this.createFromArgsRef("1", "2", "3", "4");

        test.remove("5");

        assertEquals(ref, test);
    }

    @Test
    private void removeAnyTest() {
        Set<String> test = this.createFromArgsTest("1", "2", "3", "4", "5");
        Set<String> ref = this.createFromArgsRef("1", "2", "4", "5");

        test.removeAny();

        assertEquals(ref, test);
    }

    @Test
    private void containsTest() {
        Set<String> test = this.createFromArgsTest("1", "2", "3", "4", "5");

        test.contains("2");

        assertEquals(true, test);
    }

    @Test
    private void sizeTest() {
        Set<String> test = this.createFromArgsTest("1", "2", "3", "4");

        int size = test.size();

        assertEquals(4, size);
    }
```