The PuzzleSolve algorithm in the textbook (pgs 147-148) manipulates *sets of characters*. Your exercise for today is to create a class that behaves as a set of characters.

You class should implement the following interface:

```
public class CharSet {
    public CharSet();
    public boolean contains(char c);
    public void add(char c);
    public void remove(char c);
}
```

- As defined mathematically, sets cannot contain duplicates.
- Your `CharacterSet` class should only store letters (no digits or punctuation). It should also treat lower case and capital letters as the same thing (‘R’ is the same as ‘r’). Since no letter can be duplicated, the maximum number of characters that could be in the set is 26.
- The `add()` method adds the passed character to the set, if it is not already in the set. It can ignore characters already in the set and it can ignore non-letters.
- The `remove()` method removes the passed character from the set, if it is in the set. It can ignore characters not in the set and it can ignore non-letters.
- The `contains()` function returns `true` if the passed character is in the set, otherwise it returns `false`. It should return `false` for any non-character.
- The constructor should initialize the object to be an empty set.
- The easiest way to deal with the equivalence of lower case and capital letters is to convert all characters to lower case when they are passed in to any method.

The following method may be useful:

```java
private char checkCharacter(char c) throws Exception {
    if ((c>='a')&&(c<='z')) return c;
    if ((c>='A')&&(c<='Z')) return java.lang.Character.toLowerCase(c);
    throw new Exception("invalid character");
}
```

In addition to implementing the `CharacterSet` class, you should also create a program to test its functionality.