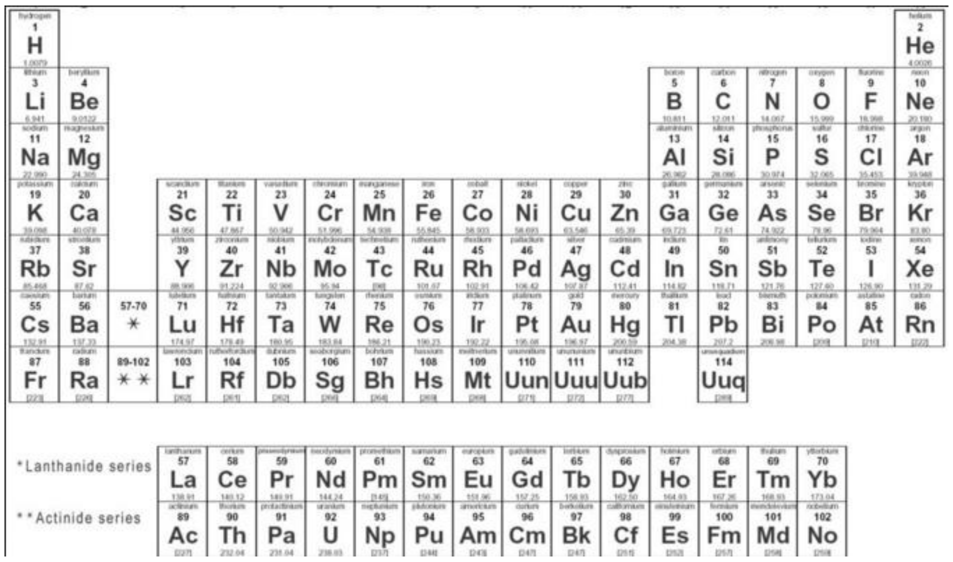
Periodic Words

This problem concerns two different tasks using the Periodic table of elements shown below.



To complete this task you have been given the completed `Element` class representing the Periodic Table and stores just the element symbol and number. Each element symbol is change to all Capitol letters. That is, He becomes HE, Uub becomes UUB and Al becomes AL.

```
public class Element
{
    private String symbol;
    private int number;

    public Element(String s, int n)
    {
        symbol = s.toUpperCase();
        number = n;
    }

    public String getSymbol()
    {
        return symbol;
    }

    public int getAtomicNumber()
    {
        return number;
    }

    public boolean equals(Object obj)
    {
        Element e = (Element)obj;
        return symbol.equals(e.getSymbol()) && number == e.getAtomicNumber();
    }

    public int hashCode()
    {
        Integer n = new Integer(number);
        return symbol.hashCode() + n.hashCode();
    }
}
```

The following table shows results of the `Element` class and its methods.

The following code	Returns

`Element e1 = new Element("He", 2));`
`Element e2 = new Element("Uub", 112));`
`Element e3 = new Element("Al", 13));`	

`e1.getSymbol();`	"HE"
`e2.getSymbol();`	"UUB"
`E3.getSymbol();`	"AL"

- Special note: after reviewing this problem, it became apparent that `getNumber()` method is not used in this problem. However, it was too late to remove it from the problem.

In this problem you will complete the unrelated methods `getMissingElements()` and `isPeriodicSpellingPossible` in the `PeriodicWords class`.

There are two constructors in the `PeriodicWords` class. One constructor has a single parameter. The parameter contains a list of Elements representing the elements currently in stock and is stored in the List `inStock`. Note that if there are two of the same element in stock, the element is contained twice in the List. In general, an element is listed `n` times indicating there are `n` of that of that elements in stock. The second constructor has no parameters and creates the array `elements` and fills it with all elements in the periodic table. That is, the array `elements` contains every element in the periodic table. In both constructors, elements are constructed from upper case and lower case letters as shown on Periodic table, however the Element constructor changes all letters to upper case.
*** This implies you will be working with Upper case letters only.

The `getMissingElements()` method has a single parameter `mix`, a List of Elements required to complete a project, and returns a List of the elements not currently in stock. The `getMissingElements()` method returns a List of all elements in mix that are not in the List `inStock`. Similar to `inStock`, elements are repeated to indicate the quantity needed and the elements are listed in random order. Remember, no Elements are added or removed from `mix` or `inStock`.

The following table show sample results of the `getMissingElements`.

The following code	Returns

`List<Element> inStock = new ArrayList<Element>();`
`inStock.add( new Element("H", 1));`
`inStock.add( new Element("He", 2));`
`inStock.add( new Element("O", 8));`
`inStock.add( new Element("C", 6));`
 	
`PeriodicWords pw = new PeriodicWords(inStock);`	

`List<Element>  mix = new ArrayList<Element>();`
`mix.add( new Element("H", 1));`
`mix.add( new Element("Li", 3));`
`mix.add( new Element("O", 8));`
 	
`List<Element> missing = pw.getMissingElements(mix);`	

`missing.size();`	1
`missing.get(0);`	`new Element("Li", 3)`

The `isPeriodicSpellingPossible()` method has a single `String` parameter `name` containing Upper Case letters. The `isPeriodicSpellingPossible()` method returns a true if it is possible to spell (create a String) `name` using only the symbols found in the Period table.
For example, BACON can be spelled (using the Symbols: Ba-C-O-N) while both TEST (there is a TE, ES and S, there is no T, E, or St) and COMPUTER cannot be spelled.
You may assume `name` will be given in ALL CAPITAL letters and `name.length() > 0`.

The following table show sample results of the `isPeriodicSpellingPossible`.

The following code	Returns
`PeriodicWords pw = new PeriodicWords();`	
`pw.isPeriodicSpellingPossible("BACON");`	`true`
`pw.isPeriodicSpellingPossible("HER");`	`true`
`pw.isPeriodicSpellingPossible("HERE");`	`true`
`pw.isPeriodicSpellingPossible("SPOONS");`	`true`
`pw.isPeriodicSpellingPossible("BYTE");`	`true`
`pw.isPeriodicSpellingPossible("LIKES");`	`true`
`pw.isPeriodicSpellingPossible("TEST");`	`false`
`pw.isPeriodicSpellingPossible("COMPUTER");`	`false`