Baby Names

A new fad has arisen in the area of baby names. Parents no longer prefer to call their children standard, meaningless names. Rather, they now commonly combine both of their names to form one for the child which will be perfect and unique. The parents combine their names in the following way:

1. Both the expectant mother and father write their names out on paper.
2. Next, the parents decide whose name will be the beginning and whose will be the ending.
3. Then, they take the name for the beginning and cross out letters from the back of that name. At least one letter will be crossed out and at least one letter will be left. The same crossing out of letters is done for the name being used for the ending but from the front this time. However, now when crossing out the letters, if the beginning part ends in a vowel, the ending part must start with a consonant (Y is considered a consonant, not a vowel); similarly, if the beginning part ends in a consonant, the ending part must start with a vowel. Furthermore, neither the beginning nor the ending should be left with no letters.
4. Finally, the shortened ending is appended to the shortened beginning and a potential baby name has now been created.

An example of the parents doing this process can be seen below (note: this is just one of many ways the parents could combine their two names to form a baby name).

1. Write names out
    a. Father's name: JAMES
    b. Mother's name: MARY
2. Decide ordering
    a. First half: JAMES
    b. Latter half: MARY
3. Shorten the names
    a. Shortened first half: JA
    b. Shortened second half: Y
4. Combine the names
    a. Baby name: JAY

The Problem:

In this problem you are to complete the `getBabyNames` method in the `BabyNames` class.

```
public class BabyNames {
/*
 *    firstName and secondName will contain only UPPERRCASE letters
 *    2 <= firstName.length()  <= 20
 *    2 <= secondName.length() <= 20
 */
   public List<String> getBabyNames( String firstName, String secondName)
   {
      List<String> ans = new ArrayList<String>();

      return ans;
   }
}
```

The following tables show sample results of the `getBabyNames` method.

The following code	Returns
`BabyNames bn = new BabyNames();` `List<String> ans = bn.getBabyNames("JAMES", "MARY");`	
`ans.size()`	5
`ans.get(0)`	"JAMARY"
`ans.get(1)`	"JAMERY"
`ans.get(2)`	"JAMEY"
`ans.get(3)`	"JARY"
`ans.get(4)`	"JAY"

The following code	Returns
`BabyNames bn = new BabyNames();` `List<String> ans = bn.getBabyNames("MARY", "JAMES");`	
`ans.size()`	5
`ans.get(0)`	"MAMES"
`ans.get(1)`	"MARAMES"
`ans.get(2)`	"MARES"
`ans.get(3)`	"MAS"
`ans.get(4)`	"MES"

The following code	Returns
`BabyNames bn = new BabyNames();` `List<String> ans = bn.getBabyNames("ABE", "JO");`	
`ans.size()`	1
`ans.get(0)`	"ABO"

The following code	Returns

`BabyNames bn = new BabyNames();`
 	
`List<String> ans = bn.getBabyNames("JO", "ABE");`	

`ans.size()`	1
`ans.get(0)`	"JE"