

International

3. This question involves generating a list of possible nicknames from a person's name. For the purposes of this question, a person's name contains a first (given) name and a last (family) name. You may assume that all punctuation has been removed so that names contain only letters and that all letters are in uppercase.

A partial declaration of the `NicknameGenerator` class is shown below. You will write two methods of the `NicknameGenerator` class.

```
public class NicknameGenerator
{
    /** The person's first name in all uppercase letters, initialized
     * by the constructor.
     */
    private String firstName;

    /** The person's last name in all uppercase letters, initialized
     * by the constructor.
     */
    private String lastName;

    // Constructor not shown

    /** Returns the number of vowels in lastName. */
    private int numVowels()
    { /* implementation not shown */ }

    /** Returns the index of the first vowel in lastName.
     * Returns -1 if there are no vowels in lastName.
     */
    private int indexOfFirstVowel()
    { /* implementation not shown */ }

    /** Returns a list of shortened last names, as described in
     * part (a).
     */
    public ArrayList<String> shortLastNames()
    { /* to be implemented in part (a) */ }

    /** Returns a list of nicknames, as described in part (b).
     */
    public ArrayList<String> nicknames()
    { /* to be implemented in part (b) */ }
}
```

International

Write the method `shortLastNames`, which returns a list of the shortened forms of a last name, according to the rules below. Each shortened last name must appear exactly once in the list.

The following rules are used to produce the list of shortened last names.

- If the last name contains fewer than two vowels, the list contains the complete last name as its only element.
- If the last name contains two or more vowels, the first shortened last name in the list contains all characters from the original last name up to and including the first vowel. Each subsequent shortened last name is produced by adding one additional character from the original last name. The list ends with the complete last name.

The `NicknameGenerator` class provides two helper methods. The method `numVowels` returns the number of vowels in `lastName`. The method `indexOfFirstVowel` returns the index of the first vowel in `lastName` or `-1` if there are no vowels in `lastName`.

In the table below, the vowels in each last name are underlined.

Last name	List of shortened last names
NG	["NG"]
SM <u>I</u> TH	["SMITH"]
L <u>O</u> <u>P</u> <u>E</u> S	["LO", "LOP", "LOPE", "LOPES"]
<u>A</u> S <u>H</u> E	["A", "AS", "ASH", "ASHE"]

Complete method `shortLastNames`. The helper methods `numVowels` and `indexOfFirstVowel` have been provided for you.

```
/** Returns a list of shortened last names, as described
 * in part (a).
 */
public ArrayList<String> shortLastNames()
```

International

Write the method `nicknames`, which returns a list of all possible nicknames. Each nickname must appear exactly once in the list.

A nickname is generated with the first letter of the first name, followed by "-", followed by one of the possible shortened forms of the last name. The table below shows several names and the list of nicknames that are generated from the name.

Name	List of nicknames
CELESTE NG	["C-NG"]
GLEN SMITH	["G-SMITH"]
JUANITA LOPES	["J-LO", "J-LOP", "J-LOPE", "J-LOPES"]
MARY ASHE	["M-A", "M-AS", "M-ASH", "M-ASHE"]

Assume that `shortLastNames` works as specified, regardless of what you wrote in part (a). You must use `shortLastNames` appropriately to receive full credit.

Complete method `nicknames`.

```
/** Returns a list of nicknames, as described in part (b). */  
public ArrayList<String> nicknames()
```