CS165 Worksheet on Iterable and Iterator Interfaces

Fill in the blanks to show how the following methods are implemented for the Iterable and Iterator interfaces. Also fill in the code inside the main method to show how to use the Iterable and Iterator interfaces.

```
import java.util.Iterator;
import java.util.NoSuchElementException;

/**
 *
 * This program create an Iterable array of Strings.
 * The iterator provides the next() method that returns
 * the next string in the array that starts with a vowel.
 * hasNext() checks if there is a string remaining that
 * starts with a vowel.
 *
 */

public class StringIterable implements Iterable<String> {

	private String[] data; // stores the data over which we are iterating.


	/**
	 *
	 * @param data is an array of String used to initialize the data
	 * inside the Iterable object.
	 */
	public StringIterable(String[] data) {
		this.data = new String[data.length];
		for (int i=0; i<data.length; i++)
			this.data[i]=data[i];
	}

	/**
	 * Returns the iterator representation of the data
	 */
	@Override
	public Iterator<String> iterator() {
		_____________________________________
		_____________________________________
	}

	private class StringIterator implements Iterator<String> {

		private int n = -1;    // stores the index of the next String
		// in the array that starts with a vowel.
		// Initially we haven't seen any element, so -1 is the initial value.
		// Later, if n becomes equal to array length,
		// then there is no such String left.

		/**
		 * Sets the value of n to point the first index of a String
		 * that starts with a vowel.
		 */
		public StringIterator (){
			findNext();    // find the first index where the string
					// has the necessary property.
		}
```

```
	/**
	 * finds the next value of n where the element is a String
	 * starting with a vowel
	 */
	private void findNext() {
		n++;
		while (n < data.length && !isVowelString(data[n]))
			n++;
	}

	/**
	 * @param s: String
	 * @return true if the first character of s is a vowel.
	 * Otherwise return false.
	 * Returns false if s is null or empty.
	 */
	private boolean isVowelString(String s) {
		if (s!=null)
			if(s.length() > 1) {
				switch(s.charAt(0)) {

				case 'a': case 'A': case 'e': case 'E': case 'i':
				case 'I': case 'o': case 'O': case 'u': case 'U':
				return true;

				default: return false;
				}
			}
		return false;
	}

	/**
	 * Checks if there is a String that starts with a vowel.
	 */
	@Override
	public boolean hasNext(){
		_____________________________________
	}

	/**
	 * Returns the next String that starts with a vowel.
	 * If no such String exists, then throws exception.
	 * If a string is returned, then n is moved to the next
	 * index where a String with a vowel might exist.
	 */
	@Override
	public String next() {
		____________________
		____________________
		____________________
		____________________
		____________________
	}

	/**
	 * remove not supported in this case
	 */
	@Override
	public void remove() {
		throw new UnsupportedOperationException();
	}
}
```

```
	public static void main(String[] args) {
		// create an initial data array
		String[] data = {"Albert", "elisa", "kristina", "Debbie", "craig", "Ben",
"Olivia", "Tess"};

		// generate an iterable and an iterator
		Iterable<String> names = ____________________
		Iterator<String> namesIterator = ____________________

		// use Iterator namesIterator
		System.out.println("Iterator:");
		// prints all the Strings that start with a vowel.
		____________________
		____________________
		____________________

		System.out.println();

		// Use Iterable names
		System.out.println("Iterable:");
		// prints all the Strings that start with a vowel.
		____________________
		____________________
		____________________

	}
}
```