

2. Consider the following incomplete `StringUtil` class declaration. You will write implementations for the two methods listed in this class. Information about the `Person` class used in the `replaceNameNickname` method will be presented in part (b).

```
public class StringUtil
{
    /** @param str a String with length > 0
     * @param oldstr a String
     * @param newstr a String
     * @return a new String in which all occurrences of the substring
     *         oldstr in str are replaced by the substring newstr
     */
    public static String apcsReplaceAll(String str,
                                       String oldStr,
                                       String newStr)
    { /* to be implemented in part (a) */ }

    /** @param str a String
     * @param people a list of references to Person objects
     * @return a copy of str modified so that each occurrence of a first
     *         name in people is replaced by the corresponding nickname
     */
    public static String replaceNameNickname(String str,
                                             List<Person>
                                             people)
    { /* to be implemented in part (b) */ }

    // There may be methods that are not shown.
}
```

- (a) Write the `StringUtil` method `apcsReplaceAll`, which examines a given `String` and replaces all occurrences of a designated substring with another specified substring. In writing your solution, you may NOT use the `replace`, `replaceAll`, or `replaceFirst` methods in the Java `String` class.

The following table shows several examples of the result of calling `StringUtil.apcsReplaceAll(str, oldstr, newstr)`.

Sample Questions for **Computer Science A**

str	oldstr	newstr	String returned	Comment
"to be or not to be"	"to"	"2"	"2 be or not 2 be"	Each occurrence of "to" in the original string has been replaced by "2"
"advanced calculus"	"math"	"science"	"advanced calculus"	No change, because the string "math" was not in the original string
"gogogo"	"go"	"gone"	"gonegonegone"	Each occurrence of "go" in the original string has been replaced by "gone"
"aaaaa"	"aaa"	"b"	"baa"	The first occurrence of "aaa" in the original string has been replaced by "b"

Complete method `apcsReplaceAll` below.

```

/** @param str a String with length > 0
 * @param oldstr a String
 * @param newstr a String
 * @ return a new String in which all occurrences of the substring
 *         oldstr in str are replaced by the substring newstr
 */
public static String apcsReplaceAll(String str,
                                   String oldStr,
                                   String newStr)

```

- (b) The following `Person` class contains information that includes a first (given) name and a nickname for the person.

```
public class Person
{
    /** @return the first name of this Person */
    public String getFirstName()
    { /* implementation not shown */ }

    /** @return the nickname of this Person */
    public String getNickname()
    { /* implementation not shown */ }

    // There may be instance variables, constructors, and methods not shown.
}
```

Write the `StringUtil` method `replaceNameNickname`, which takes a string and a list of `Person` objects that contain first names and a corresponding nicknames. The method is to replace all names by their nicknames in the given string. The list of `Person` objects is processed in order from lowest index to highest index. In writing your solution, you may NOT use the `replace`, `replaceAll`, or `replaceFirst` methods in the Java `String` class.

For example, assume the following table represents the data contained in the list `people`.

	<code>getFirstName()</code>	<code>getNickname()</code>
0	"Henry"	"Hank"
1	"Elizabeth"	"Liz"
2	"John"	"Jack"
3	"Margaret"	"Peggy"

Assume also that `String str` represents the following string.

"After Henry drove Elizabeth to dinner in Johnson City, Henry paid for an appetizer and Elizabeth paid for dessert."

The call `StringUtil.replaceNameNickname(str, people)` should return the following string:

"After Hank drove Liz to dinner in Jackson City, Hank paid for an appetizer and Liz paid for dessert."

Sample Questions for **Computer Science A**

In writing your solution, you must use the method `apcsReplaceAll` specified in the `StringUtil` class. Assume that `apcsReplaceAll` works as specified, regardless of what you wrote in part (a).

Complete method `replaceNameNickname` below.

```
/** @param str a String
 *   @param people a list of references to Person objects
 *   @return a copy of str modified so that each occurrence of a first
 *           name in people is replaced by the corresponding nickname
 */
public static String replaceNameNickname(String str,
                                         List<Person> people)
```

4. This question involves manipulation of one-dimensional and two-dimensional arrays. In part (a), you will write a method to shift the elements of a one-dimensional array. In parts (b) and (c), you will write methods to shift the elements of a two-dimensional array.

- (a) Consider the following incomplete `ArrayUtil` class, which contains a static `shiftArray` method.

```
public class ArrayUtil
{
    /** Shifts each array element to the next higher index, discarding the
     * original last element, and inserts the new number at the front.
     * @param arr the array to manipulate
     *     Precondition: arr.length > 0
     * @param num the new number to insert at the front of arr
     *     Postcondition: The original elements of arr have been shifted to
     *         the next higher index, and arr[0] == num.
     *         The original element at the highest index has been
     *         discarded.
     */
    public static void shiftArray(int[] arr, int num)
    { /* to be implemented in part (a) */ }

    // There may be methods that are not shown.
}
```

Write the `ArrayUtil` method `shiftArray`. This method stores the integer `num` at the front of the array `arr` after shifting each of the original elements to the position with the next higher index. The element originally at the highest index is lost.

For example, if `arr` is the array {11, 12, 13, 14, 15} and `num` is 27, the call to `shiftArray` changes `arr` as shown below.

<u>Before call</u>	0	1	2	3	4
arr:	11	12	13	14	15
<u>After call</u>	0	1	2	3	4
arr:	27	11	12	13	14

Sample Questions for **Computer Science A**

Complete method `shiftArray` below.

```
/** Shifts each array element to the next higher index, discarding the
 * original last element, and inserts the new number at the front.
 * @param arr the array to manipulate
 * Precondition: arr.length > 0
 * @Param num the new number to insert at the front of arr
 * Postcondition: The original elements of arr have been shifted to
 * the next higher index, and arr[0] == num.
 * The original element at the highest index has been
 * discarded.
 */
public static void shiftArray(int[] arr, int num)
```

- (b) Consider the following incomplete `NumberMatrix` class, which represents a two-dimensional matrix of integers. Assume that the matrix contains at least one integer.

```
public class NumberMatrix
{
    private int[][] matrix;

    /** Constructs a number matrix. */
    public NumberMatrix(int[][] m)
    { matrix = m; }

    /** Shifts each matrix element to the next position in row-major order
     * and inserts the new number at the front. The last element in the last
     * row is discarded.
     * @param num the new number to insert at the front of matrix
     * Postcondition: The original elements of matrix have been shifted to
     * the next higher position in row-major order, and
     * matrix[0][0] == num.
     * The original last element in the last row is discarded.
     */
    public void shiftMatrix(int num)
    { /* to be implemented in part (b) */ }

    /** Rotates each matrix element to the next higher position in row-major
     * order.
     * Postcondition: The original elements of matrix have been shifted
     * to the next higher position in row-major order, and
     * matrix[0][0] == the original last element.
     */
    public void rotateMatrix()
    { /* to be implemented in part (c) */ }

    // There may be instance variables, constructors, and methods that are not
    // shown.
}
```

Write the `NumberMatrix` method `shiftMatrix`. This method stores a new value `num` into the two-dimensional array `matrix` after shifting the elements to the next higher position in row-major order. The element originally at the last position in row-major order is lost.

Sample Questions for **Computer Science A**

For example, if `m1` is a reference to a `NumberMatrix` object, then the call `m1.shiftMatrix(48)` will change the values in `matrix` as shown below.

		<u>Before call</u>		
		0	1	2
0	13	14	15	
1	16	17	18	
2	19	20	21	
3	22	23	24	

		<u>After call</u>		
		0	1	2
0	48	13	14	
1	15	16	17	
2	18	19	20	
3	21	22	23	

In writing `shiftMatrix`, you must call the `shiftArray` method in part (a). Assume that `shiftArray` works correctly regardless of what you wrote in part (a).

Complete method `shiftMatrix` below.

```

/** Shifts each matrix element to the next position in row-major order
 * and inserts the new number at the front. The last element in the last
 * row is discarded.
 * @param num the new number to insert at the front of matrix
 * Postcondition: The original elements of matrix have been shifted
 * to the next higher position in row-major order, and
 * matrix[0][0] == num.
 * The original last element in the last row is discarded.
 */
public void shiftMatrix(int num)

```

- (c) Write the `NumberMatrix` method `rotateMatrix`. This method rotates all the elements to the next position in row-major order. The element originally at the last position is stored in the first position of the matrix.

In writing `rotateMatrix`, you must call the `shiftMatrix` method in part (b). Assume that `shiftMatrix` works correctly regardless of what you wrote in part (b).

Complete method `rotateMatrix` below.

```

/** Rotates each matrix element to the next higher position in row-major
 * order.
 * Postcondition: The original elements of matrix have been shifted to
 * the next higher position in row-major order, and
 * matrix[0][0] == the original last element.
 */
public void rotateMatrix()

```