✓ Which of the following statements about implementing ArrayLists are TRUE? Select TWO. 1/1

- [ ] All existing elements must be shifted when adding a new element to the end of an ArrayList.
- [x] All existing elements must be shifted when adding a new element at index 0. ✓
- [ ] No shifting of elements occurs when removing the first element in an ArrayList.
- [x] No shifting of elements occurs when removing the last element in an ArrayList. ✓

✓ The following sequence of numbers are inserted into 5 data structures under the Java Collections Framework. 1/1

20, 40, 10, 50, 30, 40

Printing the contents of the data structures gives the following output:

A: [20, 40, 10, 50, 30, 40]

B: [50, 20, 40, 10, 30]

C: [20, 40, 10, 50, 30]

D: [40, 20, 10, 30, 40, 50]

E: [10, 20, 30, 40, 40, 50]

Which of the following statements are TRUE? Choose ALL that apply.

- [ ] D can be a HashSet.
- [x] C can be a LinkedHashSet. ✓
- [ ] E can be a TreeSet.
- [x] A can be a list. ✓
- [ ] B can be a TreeSet.

✗ Q2 0/1

Which of the following statements are TRUE? Choose ALL that apply.

- [ ] Both sets and maps support accessing an element using the [] operator.
- [ ] Maps allow duplicated values.
- [x] Both sets and maps do not allow duplicated keys. ✓
- [x] Both LinkedHashSet and LinkedHashMap support access order. ✗

Correct answer

- [x] Both sets and maps do not allow duplicated keys.
- [x] Maps allow duplicated values.

✓ A MyStack class has the following declaration: 1/1

```
public class MyStack<E> {
 private ArrayList<E> list = new ArrayList<>();
 // remaining code is omitted
}
```

Which of the following statements are TRUE about MyStack? Choose TWO.

- [x] MyStack is a generic class. ✓
- [ ] E can be a primitive type or reference type.
- [ ] MyStack inherits methods from ArrayList.
- [x] MyStack is implemented using composition. ✓

✕ Choose a sequence of code from below that forms a complete remove(int index) method which removes an element in an ArrayList by a specified index. The method must throw an exception if an invalid index is provided. NO SPACE, NUMBER or SYMBOL in your answer. A sample answer with the correct format is acegik. Not all lines are used. ···/1

```
 public E remove(int index) {

a:  size--;

b:  return e;

c:  E e = data[index];

d:  data[j] = data[j+1];

e:  data[j-1] = data[j];

f:  data[size--] = null;

g:  data[--size] = null;

h:  if (index < 0 || index > size)

i:  if (index < 0 || index >= size)

j:  for (int j = index; j < size; j++)

k:  for (int j = index+1; j < size; j++)

l:  throw new IndexOutOfBoundsException ("Index: " + index + ", Size: " +
size);

 }
```

ilcjdgb ✕

Correct answer

ilckegb

✓ You are required to define a generic method named countLess that counts the number of elements in an array that is less than a specified target. 1/1

Sample output:

```
Integer[] arrayInt = {20, 50, 30, 10, 40};

String[] arrayStr = {"FCI", "FOE", "FOM", "FCA"};

System.out.println (countLess(arrayInt, 40));    // output is 3

System.out.println (countLess(arrayStr, "FCI")); // output is 1
```

Choose a sequence of code from below that forms a complete countLess method. NO SPACE, NUMBER or SYMBOL in your answer. A sample answer with the correct format is bigkopqq. Not all lines are used, and a line can be used multiple times.

```
a: public static <E> E countLess (E[] array, E target) {

b: public static <E> int countLess (E[] array, E target) {

c: public static <E extends Comparable<E>> int countLess (E[] array, E target) {

d: public static <E extends Comparable<E>> E countLess (E[] array, E target) {

e: public static <E implements Comparable<E>> int countLess (E[] array, E target) {

f: public static <E implements Comparable<E>> E countLess (E[] array, E target) {

g: E count = 0;

h: int count = 0;

i: for (E i = 0; i < array.length; i++) {

j: for (int i = 0; i < array.length; i++) {

k: if (array[i] < target) {

l: if (array[i].compareTo(target) < 0) {

m: if (array[i] > target) {

n: if (array[i].compareTo(target) > 0) {

o: count++;

p: return count;

q: }
```

chjloqqpq ✓