For all questions using a variables, a return, or loop is prohibited

Answer question from 1 to 7
first using : match, case, ::, Nil
than using any method of the List collection

1) Find the last element of a list.

Example:

```
scala> last(List(1, 1, 2, 3, 5, 8))
res0: Int = 8
```

2) Find the *K*th element of a list.

By convention, the first element in the list is element 0.

Example:

```
scala> nth(2, List(1, 1, 2, 3, 5, 8))
res0: Int = 2
```

3) Reverse a list.

Example:

```
scala> reverse(List(1, 1, 2, 3, 5, 8))
res0: List[Int] = List(8, 5, 3, 2, 1, 1)
```

4) Eliminate consecutive duplicates of list elements.

If a list contains repeated elements they should be replaced with a single copy of the element. The order of the elements should not be changed.

Example:

```
scala> compress(List('a, 'a, 'a, 'a, 'b, 'c, 'c, 'a, 'a, 'd, 'e, 'e, 'e, 'e))
res0: List[Symbol] = List('a, 'b, 'c, 'a, 'd, 'e)
```

5) Run-length encoding of a list.

Consecutive duplicates of elements are encoded as tuples `(N, E)` where *N* is the number of duplicates of the element *E*.

Example:

```
scala> encode(List('a, 'a, 'a, 'a, 'b, 'c, 'c, 'a, 'a, 'd, 'e, 'e, 'e, 'e))
res0: List[(Int, Symbol)] = List((4,'a), (1,'b), (2,'c), (2,'a), (1,'d), (4,'e))
```

6) Modified run-length encoding.

Modify the result of problem 5 in such a way that if an element has no duplicates it is simply copied into the result list. Only elements with duplicates are transferred as `(N, E)` terms.

Example:

```
scala> encodeModified(List('a, 'a, 'a, 'a, 'b, 'c, 'c, 'a, 'a, 'd, 'e, 'e, 'e, 'e))
res0: List[Any] = List((4,'a), 'b, (2,'c), (2,'a), 'd, (4,'e))
```

7) Decode a run-length encoded list.

Given a run-length code list generated as specified in problem 5, construct its uncompressed version.

Example:

```
scala> decode(List((4, "a"), (1, "b"), (2, "c"), (2, "a"), (1, "d"), (4, "e")))
res0: List[String] = List("a", "a", "a", "a", "b", "c", "c", "a", "a", "d", "e",
"e", "e", "e")
```