

Development 8 - Exercises

Unit 2

For these exercises use the polymorphic definition of `List<a>` using discriminated unions seen in class:

```
type List<a> = {  
  kind: "empty"  
} | {  
  kind: "cons"  
  head: a  
  tail: List<a>  
}
```

Exercise 1:

Implement a function

```
let last = <a>(l: List<a>): a
```

that returns the last element of a list.

Exercise 2:

Implement a function

```
let rev = <a>(l: List<a>): List<a>
```

that creates a list with the elements of `l` in reverse order.

Exercise 3:

Implement a function

```
let append = <a>(l1: List<a>) => (l2: List<a>):  
List<a>
```

that adds all the elements of `l2` after those in `l1`.

Exercise 4:

Implement a function

```
let nth = <a>(n: number) => (l: List<a>): a
```

that returns the element in position **n** in **l**.

Exercise 5:

Implement a function

```
let palindrome = <a>(l: List<a>): boolean
```

that checks if a list is palindrome. A list is palindrome if it is equal to its inverse.

Exercise 6:

Implement a function

```
let compress = <a>(l: List<a>): List<a>
```

that removes consecutive occurrences of the same element in the list. For example **compress** `[a;a;a;a;b;b;c;c;b]` = `[a;b;c;b]`.

Exercise 7:

Implement a function

```
let caesarCypher = (l: List<string>) => (shift: number):  
List<string>
```

The Caesar's cypher take a text, represented as a list of characters (note that Typescript does not support the type **char** so you can use a list of **string** with only one character), and shifts all the letters (so only if the character is an alphabetical character) in it up by the number of position specified by **shift**. If the letter goes past **z** it restarts from **a**. You can assume that all the text is in lower-case letter. For instance:

```
shift("c")(5) = h  
shift("y")(5) = d
```

The ASCII code for a specific character in a **string** can be obtained by using the method **charCodeAt** that takes as input the position of the character of the string you want to get the ASCII code of. For instance:

```
"Caesar".charCodeAt(2) = 101
```

Advanced: Try to support also upper-case letters in the text.