

- Please write all the Haskell code into a single `.hs`-file and upload it in OLAT.
- You can use the template `.hs`-file that is provided on the proseminar page.
- Your `.hs`-file should be compilable with `ghci`.
- Don't forget to mark your completed exercises in OLAT.

Exercise 9.1 *Lists***4 p.**

The exercise can completely be solved when creating and accessing lists purely via the functions given on slide 56 of part 3.

It is however also allowed to use pattern matching on lists. This topic will be illustrated in the lecture on Monday 7, cf. slides 6–13 of part 4.

1. Write a function `removeAt n` which removes the `n`-th element in a list. Remember that the indexing in a list in Haskell starts at zero. Return the list unmodified if there is no `n`-th element or if the given index is less than zero.

Examples:

```
removeAt 2 ["Innsbruck", "Tirol", "Deutschland", "Wien"] == ["Innsbruck", "Tirol", "Wien"]
removeAt 0 ["Innsbruck", "Tirol", "Deutschland", "Wien"] == ["Tirol", "Deutschland", "Wien"]
removeAt 66 [1,2,3] = [1,2,3]
```

(1 point)

2. Write a function `insertS x ys` which inserts an element `x` into the list `ys` after the first element in `ys` which is greater than or equal to `x`. If no such element exists in `ys`, append `x` at the end. The inserted element and the elements in the list must have a datatype which is an instance of `Ord`.

Examples:

```
insertS 4 [1,5,6,2] == [1,5,4,6,2]
insertS 3 [] == [3]
insertS "A" ["A","B","C"] == ["A","A","B","C"]
insertS 50 [0,0] == [0,0,50]
```

(1 point)

3. Write a function `unique` that removes all duplicates of the first element of the list from the list. The first element stays in the list. `unique []` should return the empty list.

Examples:

```
unique [1,2,2,3,1,1,4,1,5,6] == [1,2,2,3,4,5,6]
unique [] == []
unique [3,3] == [3]
```

(1 point)

4. Write a function `s1` which returns the second largest number in a list of `Integers`. Your function does not need to work for lists with less than two elements. In this exercise, if the largest number appears twice, `s1` should return the largest number (see examples).

Examples:

```
s1 [7,3,2,5,6] == 6
```

```
s1 [1,1,1] == 1
```

```
s1 [5,5,5,4] == 5
```

(1 point)