

## Exercises Week 7 (for December 2)

Study Sections 1.1 — 1.3 and 1.9 of OCaml reference manual.

1. [2+2+2 POINTS] Use recursion to implement the following functions:

- (a) `range : int -> int -> int list`
- (b) `sum : int list -> int`
- (c) `suffix : 'a list -> 'a list list`

```
# range 1 5;;
- : int list = [1; 2; 3; 4; 5]
# sum [1; 2; 3; 4];;
- : int = 10
# suffix [1; 2; 3];;
- : int list list = [[1; 2; 3]; [2; 3]; [3]; []]
```

2. [2+2 POINTS] Implement an *insertion sort* as follows.

- (a) Define `insert`:

```
# insert 1 [0; 2; 4];;
- : int list = [0; 1; 2; 4]
# insert 2 [0; 2; 4];;
- : int list = [0; 2; 2; 4]
```
- (b) Implement the insertion sort `isort`:

```
# isort [2; 3; 1];;
- : int list = [1; 2; 3]
```

### Submission

- Submit by email:

```
To: nao.hirokawa@uibk.ac.at
Subject: week7
```

- Deadline is Wednesday 23:59.
- Please attach your `MatrNr.ml` to the mail:

```
(* Name: your name *)
(* MatrNr: your MatrNr *)

(* 1(a) *)
let rec range a b =
  if ...

(* 1(b) *)
let rec sum = function
  | ...
```