

# Homework 14 - Ocaml

\* If there is any problem, please contact TA.

Name: \_\_\_\_\_ Student ID: \_\_\_\_\_ Email: \_\_\_\_\_

## Problem 1. Getting Started

To begin with, you have to set up OCaml on your computer first. You can find instruction [here](#). Please make sure that you can run OCaml in the **OCaml top level**. You may use any IDE you want, like Visual Studio Code with the OCaml Platform extension. For simple tests, you can also try interactive toplevel along with [UTop](#).

## Problems

Finish in the code in file `p1-base.ml` for the following functions:

- A function called `sum` that takes an integer `n` and returns the sum of the numbers from 1 to `n`.
- A function called `add2` that takes a tuple of ints (type `int * int`) and adds the two ints together.
- An inductive type called `tree` that is either a `Leaf` with an `int` value, or a `Node` with two subtrees (each of type `tree`).
- A function called `tree_size` that takes a `tree` and returns the total number of elements (nodes and leaves) in it.

**Remark:** Please include the modified code file `p1.ml` and a screenshot of execution result with your own test cases for each function in your submit.

## Problem 2. How to read numbers?

As non-native English speakers, we often have trouble pronouncing a number. Therefore, we try to write a program using OCaml to help us: For any input number (less or equal than 10 digits), this program will tell us how to pronounce the number with words.

Example input/output:

```
Enter the difficult number: 782134831
```

```
The number 782134831 read as: seven hundred eighty two million one hundred
thirty four thousand eight hundred thirty one
```

```
Enter the difficult number: 23
```

```
The number 23 read as: twenty three
```

**Remark:** Please include your code file and a screenshot of execution result with your own test cases in your submit.