

CSE 3302/5307 Programming Language Concepts

Homework 13 - Fall 2024

Due Date: Nov.24, 2024, 11:59p.m. Central Time

Problem 1 - 40%

```
(* question 1 *)
let rec sum (n : int) : int = (* change this *) 0;;

(* question 2 *)
let add2 (x : int * int) : int = (* change this *) 0;;

(* question 3 *)

type tree = (* change this *)..;;

(* question 4 *)
let rec tree_size (t : tree) : int = (* change this *) 0;;
```

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` (shown above) 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: The PDF you submit should include: (i), the OCaml code for these functions and (ii), a test for each of them and (iii), a screenshot for the output of the test from each of them.

Problem 2 - 60%

Consider extending the interpreter implemented in the slides to also handle exponentiation, modulo operation as well as greater than, smaller than and is equal operators.

Remark: Show each of scanner.mll and ast.mli and parser.mly and calc.ml in the submitted PDF and show seven test cases and their output where five of the test cases should be designated to test the five added operations.