

**Problem Set 10**  
**Comp 301**  
**Fall 2022**  
**Week 12: 09.12.2022 - 13.12.2022**

Please use the given code boilerplate for Call by Need and Call by Name languages in Problem 2.

You are not going to make any changes to the code, so submit your answers in a pdf file. Name it as ID\_username.pdf with your ID and username (Example: 1234567\_gsari18.pdf). You are expected to submit your work by December 31, 2022 midnight. **Read the question carefully. Good luck!**

**Problem 1:** What does the following program return if we run the program using

- (a) Call by Value?
- (b) Call by Reference?

```
let reset = proc (x) set x = 0 in
  let inc = proc (x) set x = -(x, -1) in
    let a = 0 in
      begin
        (inc a);
        (inc a);
        (reset a);
        (inc a);
        a
      end
    end
```

**Problem 2:** In the code boilerplate you are given the instrumented versions of Call by Name and Call by Need. The instrumented version allows you to see the state of the environment and store during evaluation. Evaluate the following expressions in top.scm of both languages and briefly explain the behavior.

- (1) For the programs below draw the environment and store diagram both for Call by Need and Call by Name before and after the red expressions are evaluated.
  - (a) let a=23 in let f=proc(x) begin x; x; x end in (f begin set a=-(a, -10); a end)
  - (b) let a=23 in let f=proc(x) begin x; x; x end in (f begin set a=-(a, -10); a end)
- (2) letrec infinite-loop(c)=(infinite-loop -(c, 1)) in let f= proc (x) begin -(2, 1); 10 end in (f (infinite-loop 99))
  - (a) What would happen if you ran the program above in IREF?
  - (b) Did you observe a difference between Call by Name and Call by Need when evaluating the above program, explain the reason briefly.
  - (c) For the program above draw the environment and store diagrams during (f (infinite-loop 99)) is evaluated.