

Problem Set 4
COMP301 FALL 2023
Week 4: 20.03.2023 - 24.03.2023

Instructions:

- Submit your answers to the Blackboard PS4 assignment until March 25 Saturday, at 23.59.
- Please submit only **one single PDF file**, where all of your codes for each of the parts are included.
- Name your submission file as *id_username_ps4.pdf*
(Example: *00000_edemirbas17_ps4.pdf*).

Problem 1: In the lecture you have seen a procedural implementation of the environment. Below, there is a procedural implementation of the list. Fill in the blank part.

```
(define (empty-list)
  (lambda (mode)
    (display "end_of_list")))
```

```
(define (prepend-list a lst)
  (lambda (mode)
    (-----
     -----
     -----)))
```

```
(define (car-list lst)
  (lst #t))
```

```
(define (cdr-list lst)
  (lst #f))
```

```
; Tests:
; (define x (prepend-list 13 (prepend-list 3 (prepend-list 6
; (prepend-list 7 (empty-list))))))
; (car-list x) -> returns 13
; (car-list (cdr-list x)) -> returns 3
; (car-list (cdr-list (cdr-list (cdr-list (cdr-list x))))) -> returns "end of list"
```

Problem 2: Given a list, an element and a number, implement a procedure "remove-n-times" that removes the element from the list n times. If the element does not occur n times in the list, it removes all occurrences.

```
(remove-n-times 'a '(a b a a b a) 2) ; returns (b a b a)
(remove-n-times 'a '(a b a) 3) ; returns (b)
```

Problem 3: Given a nested list and an input, implement a procedure named "count-occurrence-nested" that counts the occurrence of the given element in the nested list.

```
(count-occurrence-nested '(a b (a b (a b)) (a b)) 'a) ; returns 4
(count-occurrence-nested '(a (b a) (a b) (a b c)) 'b) ; returns 3
```