# Assignment 4-B

- Read Chapter15 Appendix about the syntax of scheme functional programming, then answer the questions in the next slides

- Extra YouTube videos
- Introduction to Scheme
- Lambdas functions and Conditionals
- Abstraction

## Q1. What does the following Scheme function do?

```
(define (y s lis)
        (cond
          ((null? lis) '() )
        ((equal? s (car lis)) lis)
          (else (y s (cdr lis)))
))
```

# Q2. What does the following Scheme function do?

```
(define (x lis)
  (cond
    ((null? lis) 0)
    ((not (list? (car lis)))
      (cond
        ((eq? (car lis) #f) (x (cdr lis)))
        (else (+ 1 (x (cdr lis))))))
    (else (+ (x (car lis)) (x (cdr lis))))))
```

# Q3. Change the following function

The following function returns the number of zeros in a given simple list of numbers

```
(define (countZero aList)
  ;; assume that aList is a list of numbers
  (cond ( (null? aList) 0 )
        ( (eq? (car aList) 0) (+ 1 (countZero (cdr aList))))
        ( #t (countZero (cdr aList)))
        )  )
```

Change this function to make it return the number of element x in simple list.

For example: number of **1** in the list '(1 2 5 4 1 3 1) is 3

For example: number of **4** in the list '(4 2 4 4 4 3 4 6 7) is 5

# Q4. Write the following function

- Write a Scheme function that takes a simple list of numbers as a parameter and returns a list with the largest and smallest numbers in the input list.

- For example: the list ‘(1 2 5 4 1 3 1) returns

  ‘(5 1)

# Q5. Write the following function

- Write a Scheme function that takes a list and an atom as parameters and returns a list identical to its parameter list except with all instances of the given atom deleted.
- Assume the function name is deleteatom, then, you can use the function as follows:

```
>  (deleteatom '2 '(2 3 1 4 5 6 -1 2 7))
'(3 1 4 5 6 -1 7)
> (deleteall 'a '(a r f t a r c d a e))
'(r f t r c d e)
```