

Programming Languages

CLISP Assignment 2

Instructions:

- All your answers must be in a single text named **<yourFAUID_CSLIP2.lisp>**
- Put all these three functions in one single file, named above
- Include your name, your email address at the beginning of the assignment.
- Your assignment must be submitted through Canvas by the due date
- No late submission is accepted
- All work must be on your own

Note: This assignment will be graded on a 100-point scale.

Answer the Following:

1. Implement a recursive function called **FILTER-OUT-THE** in Common Lisp. **FILTER-OUT-THE** takes a list of symbols and returns a list from which all instance of the symbols **THE** have been removed. The prototype of **FILTER-OUT-THE** is:

```
(defun filter-out-the (list)
```

```
  Your code goes here )
```

Sample Run:

```
[1] (filter-out-the '(There are the boy and THE girl))
```

```
Return: (THERE ARE BOY AND GIRL)
```

Note:

- a) You cannot use predefined function to do the elimination. You must implement it yourself.
 - b) The symbol “the” is case insensitive, which means you should eliminate every instance of “the” in all forms, regardless the combination of lower-case and upper-case.
 - c) The allowed CLISP constructs are: **COND**, **EQUAL** (or **EQUALP**), **CONS**, **CAR** and **CDR**, and some basic primitive building blocks of CLISP
2. Write a procedure called **SCHEDULE** takes a weekday as argument and retrieves a list of your commitments on that day. The procedure’s prototype is:

```
(defun schedule (day)
```

```
  Your code goes here )
```

Sample Run:

```
[1] >(load 'yourFAUID.lisp)
[2] >(schedule 'monday)
((VECTOR-CALCULUS "9:30-11:00") (PHYSICS "11:00-12:00") (PHYS-ED "3:00-4:00"))
[3] >(schedule 'tuesday)
((ECONOMICS "10:00-11:00") (PHYSICS-LAB "5:30-7:00"))
[4] >(schedule 'wednesday)
((VECTOR-CALCULUS "9:30-11:00") (PHYSICS "11:00-12:00") (PHYS-ED "3:00-4:00"))
[5] >(schedule 'thursday)
((ECONOMICS "10:00-11:00"))
[6] >(schedule 'friday)
((VECTOR-CALCULUS "9:30-11:00") (PHYSICS "11:00-12:00") (PHYS-ED "3:00-4:00"))
[7] >(schdule 'saturday)
nil
[8] >(schedule 'sunday)
NIL
```

3. Write a function called **eq_list** that compares two lists. If they are the same, return T, otherwise, return NIL

```
(DEFUN eq_list (list1 list2)
  ;; Your code goes here )
```

To test your function, do the following.

Sample Run:

```
[34] >(load 'yourFAUID.lisp)
;; Loading file yourFAUID.lisp ...
;; Loaded file yourFAUID.lisp
T
[35] >(eq_list '(1 2 3) '(1 2 3))
T
[36] >(eq_list '(1 2 3) '(1 2 3.0))
NIL
[37] >(eq_list '(ab cd ef) '(ab cd ef))
T
[38] >(eq_list '(ab cd ef) '(ab cd fe))
NIL
[39] >(eq_list '(abc 123) '(abc 123))
T
[40] >(eq_list '(abc 123) '(ABC 123))
T
[41] >(eq_list '(abc 123) '(ABC))
NIL
```