# LISP ASQ

Part 5

## Is the following function tail recursive?

```
(defun reverse-my (zoz)
  (cond ((null zoz) nil)
            ( t (append (reverse-my (rest zoz))
                              (list (first zoz))))        ))
```

yes

<u>no</u>

## Is the following function tail recursive?

```
(defun is-list (s-exp)
          (cond ((atom s-exp) (eq s-exp nil))
                   ( t (is-list (rest s-exp)))
  ))
```

<u>yes</u>

no

# Is the following function tail recursive?

```
(defun factorial (n)
          (cond ((= n 0) 1)
                   ((> n 0) (* n (factorial (- n 1))) )
 ))
```

yes

[✓ no]

## Is the following function tail recursive?

```
(DEFUN NUM-ONE-EL (LST)
        (COND ((NULL LST) 0)
                 ((ONE-EL (FIRST LST)) (+ 1 (NUM-ONE-EL (REST LST)))  )
                 ( T  (NUM-ONE-EL (REST LST)))   ))
```

yes

[✓ no]

## Is the following function tail recursive?

```
(DEFUN NO-NUM (SE)
   (COND ((NUMBERP SE) NIL)
                 ((ATOM SE) T)
                (T    (AND (NO-NUM (FIRST SE))
                           (NO-NUM (REST SE))))    ))
```

yes

[✓ no]

# Is the following function tail recursive?

```
(DEFUN EQUAL (SV1 SV2)
   (COND ((ATOM SV1) (EQ SV1 SV2))
              ((ATOM SV2) NIL)
               ((EQUAL (FIRST SV1) (FIRST SV2))
                                   (EQUAL (REST SV1) (REST SV2)))
               (T NIL) ))
```

yes

<u>no</u>

Define a function SUBST-NUM, which substitutes in a list of numbers all odd numbers by 0 and all even numbers by 1 on the top level in the list

```
(DEFUN SUBST-NUM (LST)
        (COND ((NULL LST) NIL)
                ((AND (NUMBERP (FIRST LST)
                        (ODDP (FIRST LST))) )
                            (CONS 0 (SUBST-NUM (REST LST)))  )
                ( T  (CONS 1 (SUBST-NUM (REST LST)))  )
))
```

Doplnit modre

How many conditions in COND form is necessary for the definition of function SUBST-NUM, which substitutes in a list of s-expressions all odd numbers by 0 and all even numbers by 1 on the top level in the list

2

3

[✓ 4]

5

Define a function SUBST-NUM, which substitutes in a list of s-expressions all odd numbers by 0 and all even numbers by 1 on the top level and deletes all other s-expressions

```
(DEFUN SUBST-NUM (LST)
          (COND ((NULL LST) NIL)
                      ((AND (NUMBERP (FIRST LST) )
                                (ODDP (FIRST LST)))
                                        (CONS 0 (SUBST-NUM (REST LST)))  )
                      ((AND (NUMBERP (FIRST LST) )
                                (EVENP (FIRST LST)))
                                       (CONS 1 (SUBST-NUM (REST LST)))  )
                      (  T (SUBST-NUM (REST LST))  )
))
```

Doplnit modre

Define a function SUBST-NUM-D, which substitutes in a list all odd numbers by 0 and all even numbers by 1 on the arbitrary level in the list, we suppose that the list contains on arbitrary level just numbers

```
(DEFUN SUBST-NUM-D (SE)
         (COND ((ODDP SE) 0)
                  ((EVENP SE) 1)
                  ((ATOM SE) SE)
                  ( T (CONS (SUBST-NUM-D (FIRST SE))   ) <- TUTO POSLEDNU ASI VYMAZAT
                              (SUBST-NUM-D (REST SE))  ))
))
```

Doplnit modre