LISP ASQ

<blockquote>

Part 3

Rewrite the list to dot pair notation
(C NIL)

(C . (NIL . NIL))

Rewrite the list to dot pair notation
(A (B C))

(A . ((B . (C. NIL) . NIL))

Which expression evaluates to the following (A B (C D))

<u>(APPEND ‘(A B) ‘((C D))</u>

(LIST ‘(A B) ‘(C D))

(CONS ‘(A B) ‘(C D))

Evaluate the following expression (FIRST ‘(((A)) (B C D E)))

((A))

Evaluate the following expression
(REST ‘(((((F))))))

NIL

Evaluate the following expression (FIRST ‘(REST (A B C)))

REST

Evaluate the following expression (FIRST (FIRST (REST (REST ‘((A B) (C D) (E F))))))

E

Evaluate the following expression
(FIRST (REST ‘((A B) (C D) (E F))))))

(C D)

Evaluate the following expression:

```
(EVAL (CONS '+ '(2 3)))
```

5

Evaluate the following expression:
(EVAL (LIST ‘REST (LIST ‘QUOTE ‘(1 2 3))))

(2 3)

Write a condition which returns T when the first element of a list LST is number:

```
(DEFUN FIRSTNUM (LST) __________ )
```

```
(NUMBERP (FIRST LST))
```

Write a condition which returns T when the list LST has less than 4 elements:

```
(DEFUN LESSTHAN4 (LST) __________ )
```

```
(NULL (REST (REST (REST LST))))
```

Implement function MY-NULL (create own implementation of NULL):

```
(DEFUN MY-NULL (SV)
        (COND (_________)
                 (T NIL)
))
```

```
(ATOM SV) (EQ SV NIL)
```

Implement function IS-LIST (create own implementation of LISTP):

```
(DEFUN IS-LIST (LST)
        (COND ((ATOM LST) (EQ LST NIL))
                 (T _________ ) ) )
```

<u>(IS-LIST (REST LST))</u>

Evaluate (VYBER-N -5 '(A B C)):

- NIL
- A
- C
- chyba

Evaluate (VYBER-N 5 '(A B C)):

- NIL
- A
- C
- chyba

Let evaluate the following sequence of expressions:

```
> (SETQ A ‘(B C D))
> (SETQ X ‘(B C D))
> (SETQ Y A)
```

Evaluate the following expressions:

```
> (EQ A X)
> (EQ A Y)
```