



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 5

Студент Лучина Е.Д

Группа ИУ7-61Б

Преподаватель Толпинская Н.Б.

Москва.
2020 г.

1. Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

```
(defun func (a) (if (eql (mod a 2) 1) (- a 1) a) )
```

2. Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

```
(defun func (a)
  (if (not (eql a 0))
      (if (< a 0) (- a 1) (+ a 1))
      ))
```

3. Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.

```
(defun func (a b)
  (
    if (< a b)
      (list a b)
      (list b a)
  ))
```

4. Написать функцию, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.

```
(defun func (a b c)
  (
    if (or (and (< a b) (< b c)) (and (< c b) (< b a)) )
    Т
    Nil ))
```

5. Каков результат вычисления следующих выражений?

```
(and 'fee 'fie 'foe) -> foe
(or 'fee 'fie 'foe) -> fee
(or nil 'fie 'foe) -> fie
(and nil 'fie 'foe) -> nil
(and (equal 'abc 'abc) 'yes) -> yes
(or (equal 'abc 'abc) 'yes) -> Т
```

6. Написать предикат, который принимает два числа-аргумента и возвращает Т, если первое число не меньше второго.

```
(defun func (a b)
  (
    if (or (> a b) (= a b))
    Т
    Nil
  )
)
```

7. Какой из следующих двух вариантов предиката ошибочен и почему?

```
(defun pred1 (x)
  (and (numberp x) (plusp x))) - верный
```

```
(defun pred2 (x)
  (and (plusp x)(numberp x))) - неверный, так как функция plusp применяется к аргументу до того, как установлено, что это число.
```

8. Решить задачу 4, используя для ее решения конструкции IF, COND, AND/OR.

```
(defun func2 (a b c)
  (
    cond
      ((< a b) (cond
                  ((< b c) T)))
      ((< c b) (cond
                  ((< b a) T)))
  )
)
```

```
(defun func1 (a b c)
  (
    if (< a b)
      (if (< b c) T)
      (if (< c b)
          (if (< b a) T))
  )
)
```

```
(defun func (a b c)
  (or (and (< a b) (< b c)) (and (< c b) (< b a)))
)
```

Вопросы

1) Определение атома, представление в памяти

Атом - базисный элемент языка lisp. В памяти представлен пятью указателями. Указателем на имя, на значение, на функцию, на список свойств, на пакет. Может обозначать символ, число, структуру, функцию.

Атомы разделяют на символьные, числовые. Есть зарезервированные атомы - T, Nil, базисные функции. Самоопределимыми атомами являются числа и строки. Числа делят на целые, вещественные, комплексные.

2) Локальное и глобальное определение значения атома

Функции set, setq, setf устанавливают значение атома глобально, связывают символ со значением, предварительно вычисляя значения аргументов

a) (set 'a 'b)

В качестве значения функция SET возвращает значение второго аргумента. Если перед первым аргументом нет апострофа, то значение будет присвоено значению этого аргумента.

b) (setq a 'b) = (set 'a 'b); q - обозначает quote

c) (setf place 'value) - в качестве первого аргумента, в отличии от предыдущих, функция может принимать не только символьный атом.

(setq x '(1 2)) x = (1 2)

(setf (car x) 5)) x = (5 2)

Функция let, let* определяют локально. И область видимости этих атомов завершается с окончанием S-выражения, в котором они определены. Обе функции принимают два аргумента: первый - список двухэлементных списков вида ((var1 value1) (var2 value2) ...). Где var - символьный атом ("имя определяемой переменной"), а value - S-выражение - значение определяемого атома.

Set работает в следующем порядке. 1) вычисляются все value 2) происходит связывание этих значений с var. В определении атомов нельзя ссылаться на описанные выше атомы, так как они еще не определены.

set* обрабатывает значения и связывает атом со значением последовательно. Поэтому в определении атомов можно использовать определенные выше атомы.

3) Eval; quote

Quote - блокировка вычислений. Используют, когда требуется не результат вычисления а само выражение. Апострофом помечается S-выражение, которое не нужно вычислять, а следует понимать как список.

(a b c) - без апострофа будет воспринято как вызов функции a с аргументами b, c. Причем в функцию будут переданы значения, которые хранятся в b и c.

(a 'b 'c) - вызов функции a с аргументами b и c.

'(a b c) - список с элементами a, b и c.

Eval - выполнение S-выражения.

