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

Тема Использование функционалов

Студент Сушина А.Д.

Группа ИУ7-61б

Оценка (баллы) _____

Преподаватель Толпинская Н.Б.

Москва.
2020 г

Цель работы: приобрести навыки использования функционалов.

Задачи работы: изучить работу и методы использования применяющих и отображающих функционалов: apply, funcall, maplist.

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

```
(defun notless(num)
        ( if (evenp num) num (+ num 1)))
```

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

```
(defun morethen(num)
        (if (< num 0) (- num 1) (+ num 1)))
```

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

```
(defun srt1(num1 num2)
        (if (< num1 num2) (list num1 num2) (list num2 num1)))

(defun srt2(num1 num2)
        (if (< num1 num2) (cons num1 (cons num2 NIL)) (cons num2 (cons num1 NIL))))
```

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

```
(defun incenter(a b c)
        (if (< b a)
            ( if (< a c)
                  T
                  NIL)
            ( if (< a c)
                  NIL
                  T)))

(defun incenter(a b c)
        (if (< b a) (< a c) (> a c)))
```

5. Каков результат вычисления следующих выражений?

a) (and 'fee 'fie 'foe) = foe
b) (or 'fee 'fie 'foe) = fee
c) (or nil 'fie 'foe) = fie
d) (and nil 'fie 'foe) = NIL
e) (and (equal 'abc 'abc) 'yes) = yes
f) (or (equal 'abc 'abc) 'yes) = T

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

```
(defun notless2 (a b)
        (not (< a b)))
> (notless2 0 1)
NIL
> (notless2 1 1)
T
> (notless2 2 1)
T
```

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

```
(defun pred1 (x)                    (defun pred2 (x)
(and (numberp x) (plusp x)))        (and (plusp x)(numberp x)))
```

Функция pred2 написана неверно, так как в случае, когда на вход функции pred2 в качестве аргумента будет подано не число, а , например, строка, произойдет ошибка в функции plusp. Если же передать в функцию pred1 в качестве аргумента строку, то сначала будет произведена проверка, что аргумент является числом, получится значение NIL и and вернет его в качестве результата.

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

```
(defun incenter(a b c)
        (if (< b a) (< a c) (> a c)))

(defun incenter (a b c)
   (cond ((< b a) (< a c)) ((< a b) (< c a))))

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

Ответы на вопросы:
1) Атом. Определение, представление в памяти
Атомы - это простейшие объекты Лиспа, из которых строятся остальные структуры.
Атомы:

- символы (идентификаторы) — синтаксически — набор литер(букв и цифр), начинающихся с буквы;
- специальные символы — {T, Nil}(используются для обозначения логических констант);
- самоопределимые атомы — натуральные числа, дробные числа, вещественные числа, строки — последовательность символов, заключенных в двойные апострофы.

Атомы в памяти представлены с помощью 5 указателей.

2) Самовычислимый атом

самоопределимые атомы — натуральные числа, дробные числа, вещественные числа, строки — последовательность символов, заключенных в двойные апострофы.

3) eval и quote

Функция EVAL возвращает результат выражения <выражение>, где <выражение> - любое выражение языка LISP. QUOTE возвращает выражение не выполняя его.

Quote и **eval** действуют во взаимно противоположенных направлениях и аннулируют эфефект друг друга.

4) Глобальные и локальные символьные атомы

Глобальные символьные атомы — значение устанавливается с помощью setf. Область видимости весь код следующий после определения.
>(setf a 9)
>a
9
Локальные значение — значение устанавливается с помощью let(let*) Область видимости является тело функции, в которой определена переменная.
>(let ((x 1) (y 2))
(+ x y))
3