

Задание 1

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

```
(defun f1(a)
  (if (evenp a)
      a
      (+ a 1)))
```

Задание 2

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

```
(defun f2(a)
  (if (< a 0)
      (- a 1)
      (+ a 1)))
```

Задание 3

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

```
(defun f3(a b)
  (if (< a b)
      (list a b)
      (list b a)))
```

Задание 4

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

```
(defun f4(a b c)
  (if (or (and (< a c) (> a b))
          (and (> a c) (< a b)))
      T
      NIL))
```

Задание 5

Каков результат вычисления следующих выражений?

(and 'fee 'fie 'foe) = FOE

(or nil 'fie 'foe) = FIE

(and (equal 'abc 'abc) 'yes) = YES

(or 'fee 'fie 'foe) = FEE

```
(and nil 'fie 'foe) = NIL
```

```
(or (equal 'abc 'abc) 'yes) = T
```

Задание 6

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

```
(defun f6(a b)
  (if (< a b)
      NIL
      T))
```

Задание 7

Какой из следующих двух вариантов предиката ошибочен и почему?

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

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

pred2 ошибочен, т.к. если x не число, то он выдаст ошибку, а pred1 просто вернёт NIL, потому что оператор AND вычисляет значения аргументов пока не встретит NIL.

Задание 8

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

```
(defun f8if(a b c)
  (if (< a c)
      (> a b)
      (and (> a c) (< a b))))
```

```
(defun f8cond(a b c)
  (cond
    ((< a c) (> a b))
    ((> a c) (< a b))))
```

```
(defun f8and_or(a b c)
  (or (and (< a c) (> a b))
      (and (> a c) (< a b))))
```

Задание 9

Переписать функцию how-alike, приведенную в лекции и использующую COND, используя только конструкции IF, AND/OR.

```
(defun how-alike(x y)
  (if (or (= x y) (equal x y))
      'the_same
      (if (and (oddp x) (oddp y))
          'both_odd
          (if (and (evenp x) (evenp y))
              'both_even
              'difference))))
```