

1 Лабораторная работа №3

Смирнов Пётр, ИУ7-65Б

1.1 Номер 1

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

```
(defun f(a)
  (cond ((oddp a) (+ 1 a))
        (t a)))

(print (f 4))
(print (f 3))
(print (f -3))
(print (f -4))
```

1.2 Номер 2

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

```
(defun f(a)
  (cond ((< a 0) (- a 1))
        (t (+ a 1))))

(print (f -4))
(print (f 4))
```

1.3 Номер 3

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

```
(defun f(a b)
  (cond ((< a b) (cons a (cons b nil)))
        (t (cons b (cons a nil)))))

)

(print (f 4 3))
(print (f 3 4))
```

1.4 Номер 4

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

```
(defun f(a b c)
  (cond ((and (<= a b) (>= a c)))
        ((and (>= a b) (<= a c)))))

(print (f 2 21 22))
(print (f 2 -10 22))
(print (f 2 21 -10))
```

1.5 Номер 5

Каков результат вычисления следующих выражений?

```
(print (and 'fee 'fie 'foe)) ;FOE
(print (or nil 'fie 'foe)) ;FIE
(print (and (equal 'abc 'abc) 'yes)) ;YES
(print (or 'fee 'fie 'foe)) ;FEE
(print (and nil 'fie 'foe)) ;NIL
(print (or (equal 'abc 'abc) 'yes)) ;T
```

1.6 Номер 6

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

```
(defun f(a b) (>= a b))

(print (f 25 23))
(print (f 23 23))
(print (f 21 23))
```

1.7 Номер 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)))

(print (pred1 '(2 3)))

; plusp - возвращает true, если число больше 0
; numberp - возвращает true, если аргумент типа число
```

Второй ошибочен, так как там проверка что это числовой атом идет после проверки на положительность.

1.8 Номер 8

Решить задачу 4, используя для её решения конструкции: только IF, только COND, только AND/OR.

```

(defun f_or (a b c)
  (or (<= b a c) (<= c a b)))

(defun f_and (a b c)
  (and (<= (min b c) a) (<= a (max b c))))

(defun f_cond(a b c)
  (cond ((<= a b) (>= a c))
        ((>= a b) (<= a c))))

(defun f_if(a b c)
  (if (<= a b) (>= a c)
      (if (>= a b) (<= a c))))

(print (f_and 2 21 22))
(print (f_and 2 -10 22))
(print (f_and 2 21 -10))

```

1.9 Номер 9

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

```

(defun my_if(x y)
  (if (= x y) 'the_same
      (if (equal x y) 'the_same
          (if (oddp x) (if (oddp y) '
both_odd 'difference)
              (if (evenp x) (if (
evenp y) 'both_even 'difference)))))))

(defun my_and_or(x y)
  (or (and (or (= x y) (equal x y)) 'the_same)
      (and (and (oddp x) (oddp y)) 'both_odd)
      (and (and (evenp x) (evenp y)) 'both_even)
      'difference)
  )

; лекционная функция
(defun how_alike(x y)
  (cond ((or (= x y) (equal x y)) 'the_same)
        ((and (oddp x) (oddp y)) 'both_odd)
        ((and (evenp x) (evenp y)) 'both_even)
        (t 'difference)))

(print (my_if 2 2))
(print (my_if 2 4))
(print (my_if 3 5))
(print (my_if 3 6))

```

```

; функция принимает 2 списка. вернуть упорядоченный по убыванию
; список из трех элементов:
;  $my\_min1^2 + my\_min2^2$ ,  $sr1 + sr2$ ,  $|max1-max2|$ 

(defun max_min (lst compfunc)
  (cond ((and (funcall compfunc (car lst) (cadr lst))
              (funcall compfunc (car lst) (caddr lst))) (car lst))
        ((funcall compfunc (cadr lst) (caddr lst)) (cadr lst))
        (t (caddr lst))))

(defun abs_of_razn (lst)
  (let ((raznost (- (car lst) (cadr lst))))
    (cond ((< raznost 0) (- raznost))
          (t raznost))))

(defun sr (lst) (/ (+ (car lst) (cadr lst) (caddr lst)) 3))
(defun kwadrat(a) (* a a))

(defun task(ar1 ar2)
  (let ((e1 (+ (kwadrat (max_min ar1 #'<=))
              (kwadrat (max_min ar2 #'<=))))
        (e2 (+ (sr ar1)
              (sr ar2)))
        (e3 (abs_of_razn (cons (max_min ar1 #'>=)
                               (cons (max_min ar2 #'>=) nil))))))
    (cond ((and (>= e1 e2) (>= e1 e3))
           (cons e1 (cond ((>= e2 e3) (cons e2 (cons e3 nil)))
                         (t (cons e3 (cons e2 nil))))))
          ((and (>= e2 e1) (>= e2 e3))
           (cons e2 (cond ((>= e1 e3) (cons e1 (cons e3 nil)))
                         (t (cons e3 (cons e1 nil))))))
          (t
           (cons e3 (cond ((>= e1 e2) (cons e1 (cons e2 nil)))
                         (t (cons e2 (cons e1 nil)))))))
    ))

(print (task '(1 2 3) '(4 5 6))) ; (17 7 3)
(print (task '(4 5 6) '(1 2 3))) ; (17 7 3)
(print (task '(0 0 1) '(2 0 8))) ; (7 11/3 0)
(print (task '(9 -1 3) '(2 4 1))) ; (6 5 2)

```