



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ _____ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Название: _____ Работа интерпретатора Lisp

Дисциплина: _____ Функциональное и логическое программирование

Студент	ИУ7-66Б	_____	А.Д. Ковель
	Группа	Подпись, дата	И. О. Фамилия

Преподаватель	_____	Н. Б. Толпинская
---------------	-------	------------------

Преподаватель	_____	Ю. В. Строганов
	Подпись, дата	И. О. Фамилия

Москва, 2023 г.

1 Практические задания

1. Чем принципиально отличаются функции *cons*, *list*, *append*?

```
1 (setf lst1 '(a b))  
2 (setf lst2 '(c d))
```

Каковы результаты вычисления следующих выражений?

```
1 (cons lst1 lst2)  
2 (list lst1 lst2)  
3 (append lst1 lst2)
```

- *cons* объединяет значения своих аргументов в точечную пару. Если вторым аргументом будет передан список, то в результате получится список, в котором второй аргумент будет добавлен в начало: ((A B)C D)
- *list* составляет из своих аргументов список: ((A B) (C D))
- *append* создает копию всех аргументов, кроме последнего, т. е. списковые ячейки. Связываются последними указателями. Результирующее значение: (A B C D)

2. Каковы результаты вычисления следующих выражений, и почему?

```
1 (reverse '(a b c))      —> Nil  
2 (reverse '(a b (c (d)))) —> Nil  
3 (reverse '(a))          —> (A)  
4 (last '(a b c))         —> (C)  
5 (last '(a))             —> (A)  
6 (last '((a b c)))        —> ((a b c))  
7 (reverse ())            —> Nil  
8 (reverse '((a b c)))     —> ((A B C))  
9 (last '(a b (c)))       —> ((c))  
10 (last ())              —> Nil
```

3. Написать, по крайней мере, два варианта функции, которая возвращает последний элемент своего списка-аргумента.

```

1      (defun make-two-list (a b)
2      (
3          if (< a b)
4              (list a b)
5              (list b a)
6      )
7      )
8      )

```

4. Написать, по крайней мере, два варианта функции, которая возвращает свой список аргумент без последнего элемента.

```

1      (defun is-first-between (first second third)
2      (
3          if (or
4              (and (> first second) (< first third))
5              (and (> first third) (< first second))
6          )
7              t
8              nil
9      )
10     )
11     )

```

5. Напишите функцию *swap-first-last*, которая переставляет в списке- аргументе первый и последний элементы.

- 1) (and 'fee 'fie 'foe) — foe
- 2) (or nil 'fie 'foe) — fie
- 3) (and (equal 'abc 'abc) 'yes) — yes
- 4) (or 'fee 'fie 'foe) — fee
- 5) (and nil 'fie 'foe) — Nil
- 6) (or (equal 'abc 'abc) 'yes) — T

6. Написать простой вариант игры в кости, в котором бросаются две правильные кости. Если сумма выпавших очков равна 7 или 11 — выигрыш, если выпало (1,1) или (6,6) — игрок имеет право снова бросить

кости, во всех остальных случаях ход переходит ко второму игроку, но запоминается сумма выпавших очков. Если второй игрок не выигрывает абсолютно, то выигрывает тот игрок, у которого больше очков. Результат игры и значения выпавших костей выводить на экран с помощью функции `print`.

```

1  (defun not-less (x y)
2    (
3      >= x y
4    )
5  )

```

7. Написать функцию, которая по своему списку-аргументу `lst` определяет является ли он палиндромом (то есть равны ли `lst` и `(reverse lst)`).

- 1) (defun pred1 (x) (and (numberp x) (plusp x)))
- 2) (defun pred2 (x) (and (plusp x)(numberp x)))

Ответ: ошибочен вариант 2. Так как `and` вычисляет аргументы до тех пор, пока не будет ясно, какой ответ надо вернуть. `pred2` при первой проверке вернет `NIL` и завершит работы не вызывая `plusp`.

8. Напишите свои необходимые функции, которые обрабатывают таблицу из 4-х точечных пар: (страна . столица), и возвращают по стране - столицу, а по столице — страну.

```

1  (defun x-from-y-to-z (x y z)
2    (if (< y x)
3      (if (< x z)
4        T Nil)
5      Nil))

```

```

1  (defun x-from-y-to-z (x y z)
2    (cond ((< y x) (cond ((< x z) T) (T Nil))) (T Nil)))

```

```

1  (defun x-from-y-to-z (x y z)
2    (and (< y x) (< x z)))

```

9. Напишите функцию, которая умножает на заданное число-аргумент первый числовой элемент списка из заданного 3-х элементного списка-

аргумента, когда а) все элементы списка — числа, б) элементы списка — любые объекты.

```
1 (defun how_alike (x y)
2   (cond ((or (= x y) (equal x y)) 'the_same)
3         ((and (oddp x) (oddp y)) 'both_odd)
4         ((and (evenp x) (evenp y)) 'both_even)
5         (T 'difference)
6   ))
```

```
1 (defun how_alike (x y)
2   (if (if (= x y)
3         (equal x y))
4       'the_same
5       (if (if (oddp x)
6               (oddp y))
7           'both_odd
8           (if (if (evenp x)
9                   (evenp y))
10              'both_even
11              'difference)
12   ))))
```

```
1 (defun how_alike (x y)
2   (or
3     (and (= x y) (equal x y) 'the_same)
4     (and (oddp x) (oddp y) 'both_odd)
5     (and (evenp x) (evenp y) 'both_even)
6     'difference
7   ))
```