



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»

Отчёт по лабораторной работе №11 по курсу «Функциональное и логическое программирование»

Тема Рекурсия на Prolog

Студент Динь Вьет Ань

Группа ИУ7И-64Б

Оценка (баллы)

Преподаватели Толпинская Н. Б., Строганов Ю. В.

Практические задания

Задание 1

Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции, проходя по верхнему уровню списковых ячеек.
(* Список смешанный структурированный).

```
1 (defun minus-ten (lst)
2   (
3     mapcar #'(lambda (el)
4       ( cond
5         ((numberp el) (- el 10)
6          (T el))
7       ) lst
8   )
9 )
```

Задание 2

Написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

```
1 (defun sqr-lst (lst)
2   (
3     mapcar #'(lambda (el)
4       (* el el)) lst
5   )
6 )
```

Задание 3

Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда: а) все элементы списка — числа, б) элементы списка — любые объекты.

а)

```
1 (defun mul-lst (lst num)
2   (
3     mapcar #'(lambda (el)
4       (* el num)) lst
5   )
6 )
```

б)

```
1 (defun mul-lst (lst num) (  
2   mapcar #'(lambda (el) (  
3     (cond ((numberp el) (* el num)  
4       (T el))  
5     ) lst  
6   )  
7 )
```

Задание 4

Написать функцию, которая по своему списку-аргументу lst определяет является ли он палиндромом (то есть равны ли lst и (reverse lst)), для одноуровневого смешанного списка.

```
1 (defun list_without_last (lst res)  
2   (cond  
3     ((null (cdr lst)) res)  
4     (t (list_without_last (cdr lst) (cons (car lst) res))))  
5 ))  
6  
7 (defun is_palindrome (lst)  
8   (cond  
9     ((null (cdr lst)) t)  
10    ((eql (car lst) (car (last lst)))  
11      (is_palindrome (list_without_last (cdr lst) ())))  
12 ))
```

Задание 5

Используя функционалы, написать предикат set-equal, который возвращает t, если два его множества-аргумента (одноуровневые списки) содержат одни и те же элементы, порядок которых не имеет значения.

```
1 (defun set-equal (set1 set2)  
2   (cond  
3     ((null set1) (null set2))  
4     ((null set2) Nil)  
5     (t (set-equal (cdr set1) (remove (car set1) set2))))  
6 ))
```

Задание 6

Напишите функцию, select-between, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию (+ 2 балла)).

```

1 (defun select-between (lst left right) (
2   cond ((< right left) Nil)
3   (T (sort (reduce #'(lambda (result el)
4             (cond ((and (> el left) (> right el))
5                   (cons el result))
6             (T result))
7     )lst :initial-value '()) #'<
8 )))

```

Задание 7

Написать функцию, вычисляющую декартово произведение двух своих списков аргументов. (Напомним, что $A \times B$ это множество всевозможных пар (a, b) , где a принадлежит A , принадлежит B .)

```

1 (defun decart (lst1 lst2) (
2   mapcan #'(lambda (x)
3     (mapcar #'(lambda (y)
4       (list x y)) lst2)) lst1
5 ))

```

Задание 8

Почему так реализовано reduce, в чем причина?

```

1 (reduce #'+ ()) -> 0
2 (reduce #'* ()) -> 1

```

Если список пуст, а начальное значение не задано, то вызывается функция без аргументов, в reduce возвращает то, что вернёт функция. Функция сложения без аргументов возвращает 0, а функция умножения возвращает 1.

Задание 9

Пусть list-of-list список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов list-of-list (количество атомов), т.е. например для аргумента $((1\ 2)\ (3\ 4)) \rightarrow 4$.

```

1 (defun len-list-of-lists (lst) (
2   apply #'+ (
3     mapcar #'(lambda (el)
4       (cond ((listp el) (len-list-of-lists el))
5       (T 1))
6     ) lst
7 )))

```