



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 № 2

Тема Определение функций пользователя

Студент Сушина А.Д.

Группа ИУ7-616

Оценка (баллы) _____

Преподаватель Толпинская Н.Б.

Москва.
2020 г.

Цель работы: приобрести навыки использования списков и стандартных функций Lisp.

Задачи работы: изучить способ использования списков для фиксации информации, внутреннее представление одноуровневых и структурированных списков, методы их обработки с использованием базовых функций Lisp.

Задание 1.

Что будет в результате вычисления выражений?

- a) (CAADR '((blue cube)(red pyramid)))
1. (CDR '((blue cube)(red pyramid))) = ((red pyramid))
 2. (CAR '((red pyramid))) = (red pyramid)
 3. (CAR '(red pyramid)) = red

Результат: red

- b) (CDAR '((abc)(def)(ghi)))
1. (CAR '((abc)(def)(ghi))) = (abc)
 2. (CDR '(abc)) = Nil

Результат: Nil

- c) (CADR '((abc)(def)(ghi)))
1. (CDR '((abc)(def)(ghi))) = ((def)(ghi))
 2. (CAR '((def)(ghi))) = (def)

Результат: (def)

- d) (CADDR '((abc)(def)(ghi)))
1. (CDR '((abc)(def)(ghi))) = ((def)(ghi))
 2. (CDR '((def)(ghi))) = ((ghi))
 3. (CAR '((ghi))) = (ghi)

Результат: (ghi)

Задание 2.

Напишите результат вычисления выражений:

- a) (list 'Fred 'and Wilma)

Результат: Error (The variable WILMA is unbound.) Wilma не определено и не является само вычисляемым, поэтому возникает ошибка.

- b) (list 'Fred '(and Wilma))

Результат: (Fred(and Wilma))

- c) (cons Nil Nil) = (Nil.Nil) = (Nil)

Результат: (Nil)

- d) (cons T Nil) = (T.Nil) = (T)

Результат: (T)

- e) (cons Nil T) = (Nil.T)

Результат: (Nil . T)

f) (list Nil) = (Nil)

Результат: (Nil)

g) (cons (T) Nil)

Результат: The function COMMON-LISP: T is undefined. Lisp интерпретирует T как функцию, а она таковой не является.

h) (list '(one two) '(free temp))

Результат: ((one two)(free temp))

i) ((cons 'Fred '(and Wilma))

Результат: (Fred and Wilma)

j) (cons 'Fred '(Wilma))

Результат: (Fred Wilma)

k) (list Nil Nil)

Результат: (Nil Nil)

l) (list T Nil)

Результат: (T Nil)

m) (list Nil T)

Результат: (Nil T)

n) (cons T (list Nil))

Результат: (T Nil)

o) (list (T) Nil)

Результат: The function COMMON-LISP: T is undefined.

p) (cons '(one two) '(free tmp))

Результат: ((one two) free tmp)

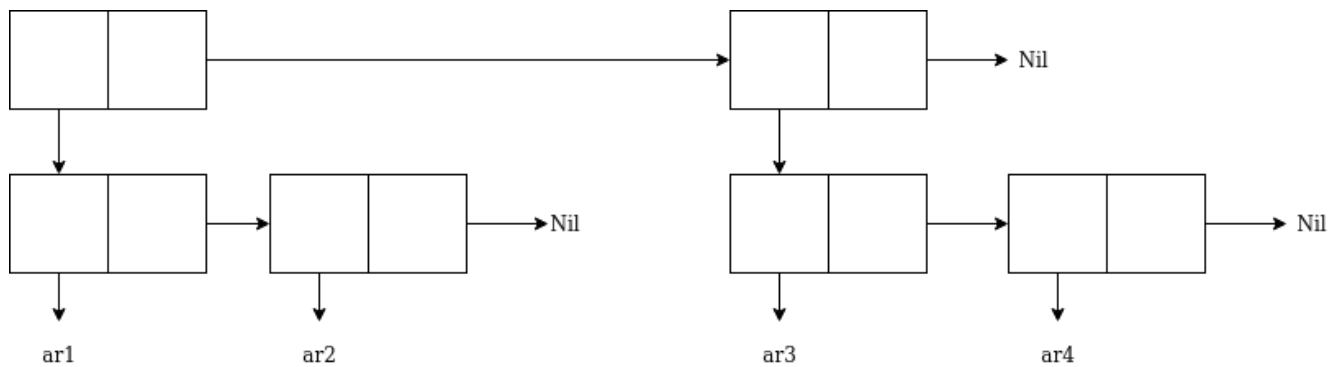
Задание 3.

a) Написать функцию (f ar1 ar2 ar3 ar4), возвращающую список: ((ar1 ar2)(ar3 ar4)).

```
(defun f1 (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
```

```
(defun f (ar1 ar2 ar3 ar4) (cons (cons ar1 (cons ar2 Nil)) (cons (cons ar3 (cons ar4 Nil)) Nil)))
```

```
(defun f (ar1 ar2 ar3 ar4) `((,ar1 ,ar2)(,ar3 , ar4)))
```

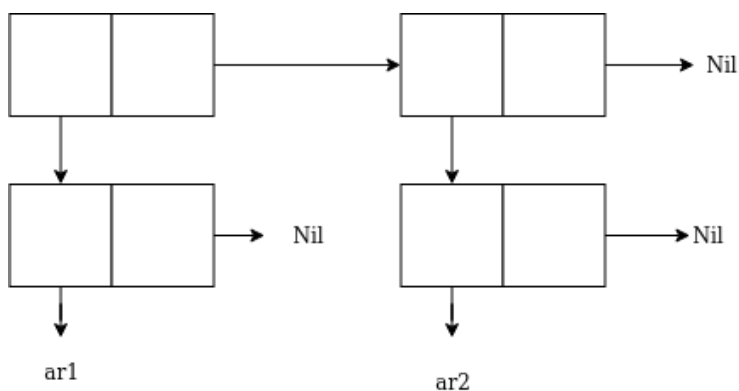


b) Написать функцию (f ar1 ar2), возвращающую ((ar1)(ar2))

```
(defun f2 (ar1 ar2) (list (list ar1) (list ar2)))
```

```
(defun f (ar1 ar2) (cons (cons ar1 Nil) (cons (cons ar2 Nil) Nil)))
```

```
(defun f (ar1 ar2) `((,ar1)(,ar2)))
```

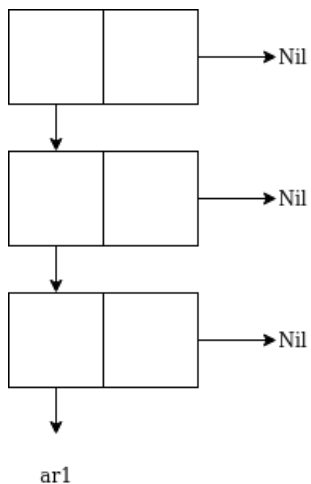


c) Написать функцию (f ar1), возвращающую (((ar1)))

```
(defun f3 (ar1) (list (list (list ar1))))
```

```
(defun f (ar1) (cons (cons (cons ar1 Nil) Nil) Nil))
```

```
(defun f (ar1) `(((,ar1))))
```



Ответы на вопросы:

1. Базис LISP

Базис – это минимально необходимый набор конструкций, с помощью которого можно реализовать задачу.

Базис Lisp образуют: атомы, структуры, базовые функции, базовые функционалы.

2. Классификация функций

- Базовые функции — принимают фиксированное количество аргументов
- Формы – принимают не фиксированное количество аргументов или обрабатывают аргументы по разному
- Функционалы – используют другие функции в качестве аргументов или вырабатывают в качестве результатов

3. Представление списков в памяти

Списочная ячейка состоит из двух частей, полей `car` и `cdr`. Каждое из полей содержит указатель. Указатель может ссылаться на другую списочную ячейку или на некоторый другой Lisp объект, как, например, атом. Указатели между ячейками образуют как бы цепочку, по которой можно из предыдущей ячейки попасть в следующую и так, наконец, до атомарных объектов. Каждый известный системе атом записан в определённом месте памяти лишь один раз.

4. CAR, CDR

CAR и CDR являются базовыми функциями доступа к данным. CAR принимает точечную пару или пустой список в качестве аргумента и возвращает первый элемент или `nil`, соответственно. CDR принимает точечную пару или пустой список и возвращает список состоящий из всех элементов, кроме первого. Если в списке меньше двух элементов, то возвращается `Nil`.