



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»

Отчет по лабораторной работе № 1 по дисциплине "Функциональное и логическое программирование"

Студент Кузнецова А. В.

Группа ИУ7-61Б

Оценка (баллы) _____

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

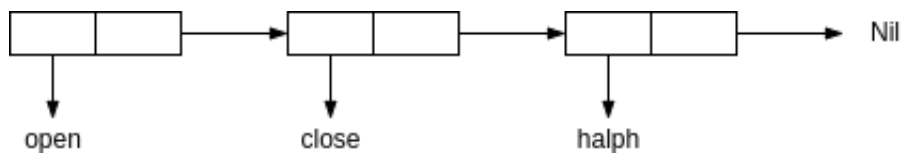
Москва — 2023 г.

1 Практические задания

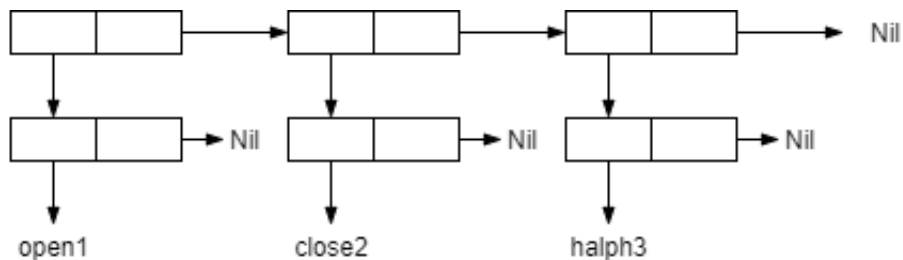
1.1 Задание №1

Представить следующие списки в виде списочных ячеек.

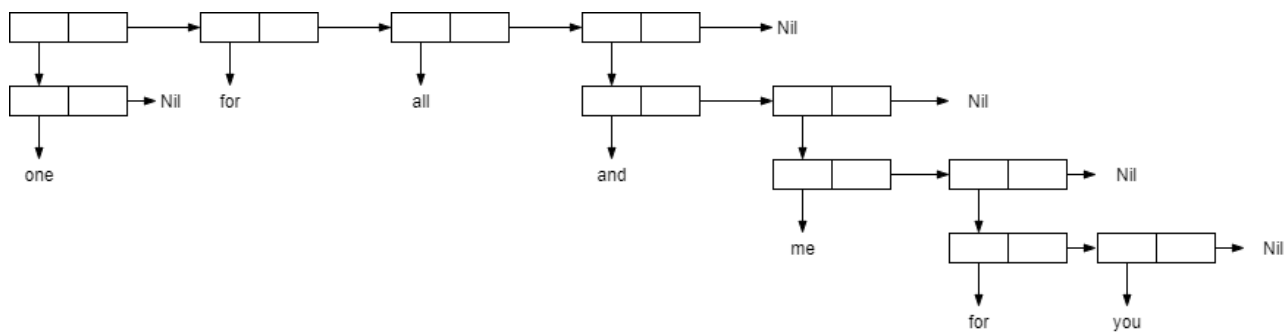
1. '(open close halph)



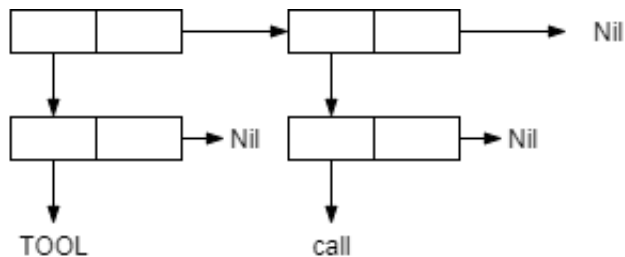
2. '((open1) (close2) (halph3))



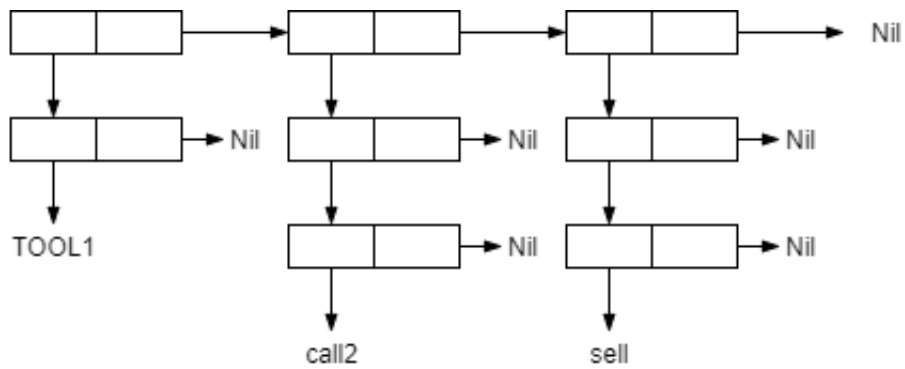
3. '((one) for all (and (me (for you))))



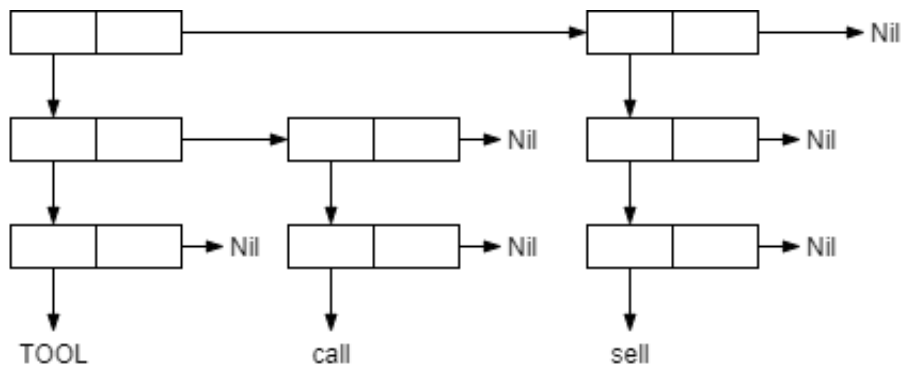
4. '((TOOL) (call))



5. '((TOOL1) ((call2)) ((sell)))



6. '(((TOOL) (call)) ((sell)))



1.2 Задание №2

Используя функции CAR и CDR, написать выражения, возвращающие:

1. второй элемент;

1

2. третий элемент;

1

3. четвертый элемент.

1

1.3 Задание №3

Что будет в результате вычисления выражений?

1. Выражение:

```
1 (caadr '((blue cube) (red pyramid)))
```

Результат: red

2. Выражение:

```
1 (cdar '((abc) (def) (ghi)))
```

Результат: Nil

3. Выражение:

```
1 (cadr '((abc) (def) (ghi)))
```

Результат: (def)

4. Выражение:

```
1 (caddr '((abc) (def) (ghi)))
```

Результат: (ghi)

1.4 Задание №4

Напишите результат вычисления выражений:

```
1 (list 'Fred 'and 'Wilma) => (Fred and Wilma).
2 (list 'Fred '(and Wilma)) => (Fred (and Wilma))
3 (cons Nil Nil) => (Nil . Nil) => (Nil)
4 (cons T Nil) => (T . Nil) => (T)
5 (cons Nil T) => (Nil . T)
6 (list Nil) => (Nil)
7 (cons '(T) Nil) => ((T) . Nil) => ((T))
8 (list '(one two) '(free temp)) => ((one two) (free temp))
9
```

```

10 (cons 'Fred '(and Willma)) => (Fred . (and Willma)) => (Fred
    and Willma)
11 (cons 'Fred '(Wilma)) => (Fred . (Willma)) => (Fred Willma)
12 (list Nil Nil) => (Nil Nil)
13 (list T Nil) => (T Nil)
14 (list Nil T) => (Nil T)
15 (cons T (list Nil)) => (cons T '(Nil)) => (T . (Nil)) => (T
    Nil)
16 (list '(T) Nil) => ((T) Nil)
17 (cons '(one two) '(free temp)) => ((one two) . (free temp))
    => ((one two) free temp)

```

1.5 Задание №5

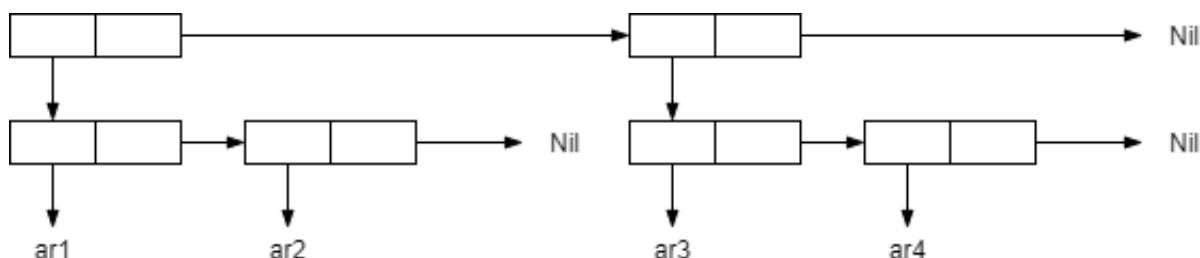
Написать лямбда-выражение и соответствующую функцию. Представить результаты в виде списочных ячеек.

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

```

1 (defun f1 (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
2
3 (lambda (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))

```

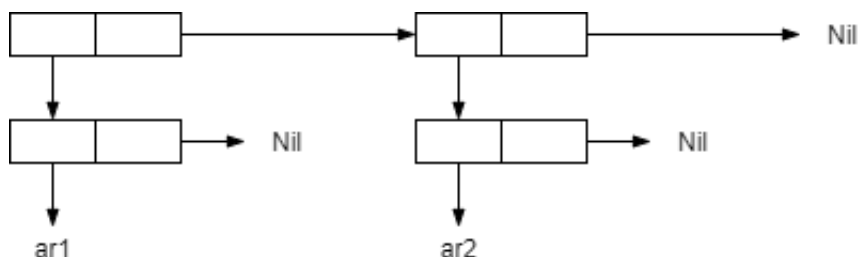


2. Написать функцию (f ar1 ar2), возвращающую ((ar1) (ar2)).

```

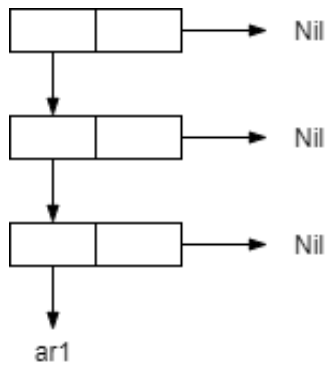
1 (defun f (ar1 ar2) (list (list ar1) (list ar2)))
2
3 (lambda (ar1 ar2) (list (list ar1) (list ar2)))

```



3. Написать функцию (f ar1), возвращающую (((ar1))).

```
1 (defun f (ar1) (list (list (list ar1))))  
2  
3 (lambda (ar1) (list (list (list ar1))))
```



2 Теоретические вопросы

2.1 Элементы языка: определение, синтаксис, представление в памяти

Вся информация (данные и программы) в Lisp представляется в виде символьных выражений — S-выражений.

S-выражение ::= <атом> | <точечная пара>

Атомы:

- символы — синтаксически — набор литер (букв и цифр), начинающийся с буквы;
- специальные символы {T, Nil} обозначают логические константы;
- самоопределимые атомы — натуральные, дробные, вещественные числа и строки.

Более сложные данные — точечные пары и списки (структуры).

Точечная пара ::= (<атом>. <атом>) | (<атом>, <точечная пара>) | (<точечная пара>. <атом>) | (<точечная пара>. <точечная пара>)

Список ::= <пустой список> | <непустой список>, где

<пустой список> ::= () | Nil,

<непустой список> ::= (<первый элемент>. <хвост>),

<первый элемент> ::= <S-выражение>,

<хвост> ::= <список>.

Синтаксически любая структура заключается в круглые скобки:

- (A . B) — точечная пара;
- (A) — список из одного элемента;
- Nil или () — пустой список;
- (A . (B . (C . (D ()))) или (A B C D) — непустой список;
- элементы списка могут являться списками: ((A)(B)(CD)).

Любая непустая структура Lisp в памяти представляется списковой ячейкой, хранящей два указателя: на голову (первый элемент) и хвост (все остальное).

2.2 Особенности языка Lisp. Структура программы. Символ апостроф

Особенности языка Lisp:

- символьная обработка данных;
- любая программа может интерпретироваться как функция с одним или несколькими аргументами;
- автоматизированное динамическое распределение памяти, которая выделяется блоками;
- бестиповый язык;
- программа может быть представлена как данные, то есть программа может изменять саму себя.

Символ апостроф — сокращенное обозначение функции `quote`, блокирующей вычисление своего аргумента.

2.3 Базис языка Lisp. Ядро языка

Базис языка — минимальный набор конструкций и структур данных, с помощью которого можно написать любую программу.

Базис Lisp образуют:

- атомы;
- структуры;
- базовые функции;

- базовые функционалы.

Функцией называется правило, по которому каждому значению одного или нескольких аргументов ставится в соответствие конкретное значение результата.

Функционалом, или функцией высшего порядка называется функция, аргументом или результатом которой является другая функция.

Помимо базиса языка, ядро включает в себя наиболее часто используемые функции, не входящие в базис.