

Федеральное агентство связи  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»

Кафедра прикладной математики и кибернетики

Лабораторная работа № 1  
по дисциплине «Функциональное и логическое программирование»

Бригада № 6

Выполнил:  
студент 3 курса  
группы ИП-811  
Мироненко К. А

Проверил:  
доцент кафедры ПМиК  
Галкина М.Ю.

## Оглавление

1. Постановка задачи.....	3
2. Примеры работы программы .....	5
<i>Приложение</i> Листинг.....	6

# 1. Постановка задачи

Каждая бригада должна выполнить 4 задачи: по одной из каждого раздела (номер задачи раздела совпадает с номером бригады).

1. Напишите сложную функцию, используя композиции функций CAR и CDR, которая возвращает атом \* при применении к следующему списку:

- 1) (((1 (2 \*)) 3) 4)
- 2) (1 (((2 3) (4 5) \*) (6)))
- 3) ((1 ((\*) 2 3)))
- 4) (1 (2 (\* 3) 4) 5)
- 5) (1 (2 (3 4) \*) 5)
- 6) ((1 2) (3 (4 \*) 5))
- 7) ((1) ((2 \*)) (3 (4)))
- 8) ((1 2 3 (4 5 (\*))))
- 9) ((1 (2 (3 4 \*))))
- 10) ((1 (2 (3 \*))))

2. Объясните работу функций и определите результат обращения:

- 1) (cons nil nil)
- 2) (list '(a b) 7 '(()))
- 3) (cons '(+ 1 2) '(+ 4 6))
- 4) (append '(+ 1 2) '())
- 5) (cons '(a (b c)) nil)
- 6) (append nil '(a b c))
- 7) (list '(+ 1 2) '() '(+ 1))
- 8) (append '(4 5 6) '())
- 9) (cons (+ 1 2) '(+ 4 6))
- 10) (list '(+ 1 2) '())

3. Из атомов 1, 2, 3, nil создайте указанные списки двумя способами:

- а) с помощью композиций функций CONS;  
б) с помощью композиций функций LIST.

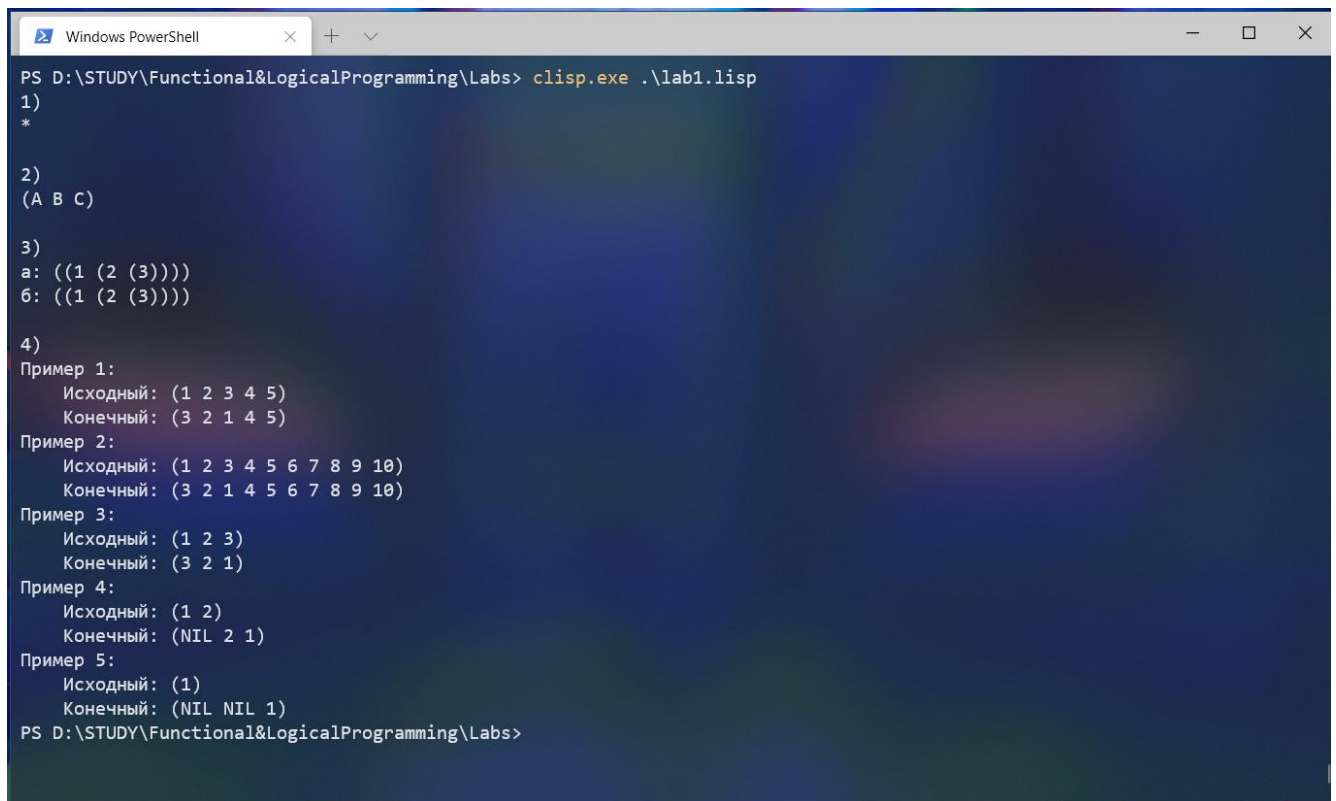
- 1) (1 (2 3))
- 2) (1 (2) 3)
- 3) (((1 2 3)))
- 4) ((1)(2 (3)))

- 5) ((1)(2 3))
- 6) ((1(2 (3))))
- 7) ((1 2 ((3))))
- 8) ((1 2) 3)
- 9) (((1)) 2) 3)
- 10) (1 ((2)) (3))

4. С помощью DEFUN определите функцию, которая возвращает измененный список по заданию (в теле функции разрешается использовать только следующие встроенные функции: CAR, CDR, CONS, APPEND, LIST, LAST, BUTLAST с одним аргументом). Проверьте её работу, организовав обращение к функции со списками разной длины.

- 1) Функция меняет местами первый и последний элементы списка
- 2) Функция меняет местами первый и второй элементы списка
- 3) Функция меняет местами первый и предпоследний элементы списка
- 4) Функция меняет местами второй и третий элементы списка
- 5) Функция меняет местами третий и последний элементы списка
- 6) Функция меняет местами первый и третий элементы списка
- 7) Функция меняет местами второй и последний элементы списка
- 8) Функция меняет местами предпоследний и последний элементы списка
- 9) Функция меняет местами второй и предпоследний элементы списка
- 10) Функция меняет местами третий и предпоследний элементы списка

## 2. Примеры работы программы



```
PS D:\STUDY\Functional&LogicalProgramming\Labs> clisp.exe .\lab1.lisp
1)
*

2)
(A B C)

3)
a: ((1 (2 (3))))
b: ((1 (2 (3))))

4)
Пример 1:
    Исходный: (1 2 3 4 5)
    Конечный: (3 2 1 4 5)
Пример 2:
    Исходный: (1 2 3 4 5 6 7 8 9 10)
    Конечный: (3 2 1 4 5 6 7 8 9 10)
Пример 3:
    Исходный: (1 2 3)
    Конечный: (3 2 1)
Пример 4:
    Исходный: (1 2)
    Конечный: (NIL 2 1)
Пример 5:
    Исходный: (1)
    Конечный: (NIL NIL 1)
PS D:\STUDY\Functional&LogicalProgramming\Labs>
```

## Приложение Листинг

```
;;; 1. Напишите сложную функцию, используя композиции функций CAR и CDR,  
;;; которая возвращает атом * при применении к следующему списку:  
;;; 6) ((1 2) (3 (4 *) 5))
```

```
;; (write (CAR (CDR (CAR (CDR (CAR (CDR '((1 2) (3 (4 *) 5))))))))  
;; (write (CADR (CADADR '((1 2) (3 (4 *) 5)))))
```

```
(format t "1)~%~A~%~%"  
  (CAR (CDR  
    (CAR (CDR  
      (CAR (CDR '((1 2) (3 (4 *) 5))  
    ))  
  ))  
))  
)
```

```
;;; 2. Объясните работу функций и определите результат обращения:  
;;; 6) (append NIL '(a b c))  
;;; Объяснение : APPEND - "склеивает" списки  
;;; Результат : (A B C)
```

```
(format t "2)~%~A~%~%" (APPEND NIL '(a b c)))
```

```
;;; 3. Из атомов 1, 2, 3, nil создайте указанные списки двумя способами:  
;;; а) с помощью композиций функций CONS;  
;;; б) с помощью композиций функций LIST.  
;;; Список: 6) ((1(2 (3))))
```

```
;; (write (CONS (CONS 1 (CONS (CONS 2 (CONS (CONS 3 ()) ()) ()) ())) ()))  
(format t "3)~%a: ~A~%b: ~A~%~%"  
  ;; а)  
  (CONS ;; Внешняя обертка  
    (CONS  
      1  
      (CONS  
        (CONS  
          2  
          (CONS  
            (CONS  
              3  
              NIL  
            )  
          )  
        )  
      )  
    )  
  )  
  NIL  
)  
)  
NIL  
)  
;; б)  
(LIST (LIST 1 (LIST 2 (LIST 3))))
```

)

;;; 4. С помощью DEFUN определите функцию, которая возвращает измененный список по заданию (в теле функции разрешается использовать только следующие встроенные функции: CAR, CDR, CONS, APPEND, LIST, LAST, BUTLAST с одним аргументом).  
;;; Проверьте её работу, организовав обращение к функции со списками разной длины.  
;;; 6) Функция меняет местами первый и третий элементы списка

```
(defun fun (x)
  ;; (CADDR x) - 3 элемент
  ;; (CDDDR x) - 4-й элемент
  ;; (CADR x) - 2 элемент
  ;; (CAR x) - 1 элемент
  (APPEND (LIST (CADDR x)) (LIST (CADR x)) (LIST (CAR x)) (CDDDR x))
)
```

```
(format t
"4)
```

Пример 1:

Исходный: ~A

Конечный: ~A

Пример 2:

Исходный: ~A

Конечный: ~A

Пример 3:

Исходный: ~A

Конечный: ~A

Пример 4:

Исходный: ~A

Конечный: ~A

Пример 5:

Исходный: ~A

Конечный: ~A"

'(1 2 3 4 5)

(fun '(1 2 3 4 5))

'(1 2 3 4 5 6 7 8 9 10)

(fun '(1 2 3 4 5 6 7 8 9 10))

'(1 2 3)

(fun '(1 2 3))

'(1 2)

(fun '(1 2))

'(1)

(fun '(1))

)