

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

Кафедра прикладной математики и кибернетики

Лабораторная работа № 2  
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 .....	6
<i>Приложение</i> Листинг.....	7

# 1. Постановка задачи

Каждая бригада должна выполнить 3 задачи (номера выбираются из таблицы). В теле функции использование операторов set, let и setq не допускается! Все функции должны быть рекурсивными, функционалы не использовать.

Номер бригады	1	2	3	4	5	6	7	8	9	10
Номера задач для бригады не более, чем из 2-х человек	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17	18	19	20
	21	22	23	24	25	26	27	28	29	30

Определите функцию:

1) Добавляющую заданный параметром символ после каждого неотрицательного элемента списка.

Например,  $x=*$ ,  $L=(-1\ d\ 6\ -3\ a\ 0) \rightarrow (-1\ d\ (6\ *)\ -3\ a\ (0\ *))$ .

2) Преобразующую список в "двойной" список таким образом, чтобы каждый элемент удвоился.

Например,  $(1\ a\ b\ 3) \rightarrow (1\ 1\ a\ a\ b\ b\ 3\ 3)$ .

3) Заменяющую в списке все вхождения  $x$  на  $y$ .

Например,  $x=1$ ,  $y=+$ ,  $L=(2\ 1\ 3\ 5\ 1\ 1\ 8) \rightarrow (2\ +\ 3\ 5\ +\ +\ 8)$ .

4) Добавляющую заданное параметром  $x$  число к каждому числовому элементу списка.

Например,  $x=3$ ,  $L=(a\ -1\ 6\ v\ 3) \rightarrow (a\ 2\ 9\ v\ 6)$ .

5) Удаляющую  $n$  первых элементов из списка.

Например,  $n=3$ ,  $L=(2\ 6\ 1\ 7\ 0) \rightarrow (7\ 0)$ .

6) Строящую список "луковица" с уровнем вложенности  $n$  для параметра  $x$ .

Например,  $n=4$ ,  $x=* \rightarrow ((((*))))$ .

7) Добавляющую заданное параметром  $x$  число в упорядоченный по неубыванию числовой список таким образом, чтобы сохранилась упорядоченность. Сортировку не использовать!

Например,  $x=7$ ,  $L=(0\ 3\ 3\ 6\ 9) \rightarrow (0\ 3\ 3\ 6\ 7\ 9)$ .

8) Возвращающую список позиций элемента, заданного параметром  $x$ , в списке  $L$ .

Например,  $x=4, L=(8\ 3\ 4\ 6\ 4\ 1) \rightarrow (3\ 5)$ .

9) Возвращающую список из  $n$  копий заданного атома  $x$ .

Например,  $x=4, n=5 \rightarrow (4\ 4\ 4\ 4\ 4)$ .

10) Удаляющую все отрицательные элементы из списка.

Например,  $(-2\ 6\ s\ -1\ 4\ f\ 0) \rightarrow (6\ s\ 4\ f\ 0)$ .

11) Объединяющую 2 списка в один, чередуя элементы списков.

Например,  $L1=(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8), L2=(a\ s\ d\ f) \rightarrow (1\ a\ 2\ s\ 3\ d\ 4\ f\ 5\ 6\ 7\ 8)$ .

12) Возвращающую список из пар соседних элементов, одинаково отстоящих от начала и конца списка.

Например,  $(1\ a\ b\ c\ 2\ 3\ f) \rightarrow ((1\ f)\ (a\ 3)\ (b\ 2)\ (c))$

13) Определяющую, сколько раз заданное  $s$ -выражение входит в список.

Например,  $x=(a), L=(1\ (a)\ x\ (a)\ 2\ a\ 1\ 2\ d) \rightarrow 2$ .

14) Преобразующую список в множество (для повторяющихся элементов должно оставаться последнее вхождение в список).

Например,  $(a\ b\ a\ a\ c\ c) \rightarrow (b\ a\ c)$ .

15) Осуществляющую циклический сдвиг в списке  $L$  на  $n$  элементов влево.

Например,  $L = (a\ s\ d\ f\ g), n = 3 \rightarrow (f\ g\ a\ s\ d)$ .

16) Осуществляющую циклический сдвиг в списке  $L$  на  $n$  элементов вправо.

Например,  $L = (a\ s\ d\ f\ g), n = 3 \rightarrow (d\ f\ g\ a\ s)$ .

17) Удаляющую элементы с четными номерами из списка (нумерация элементов должна начинаться с 1). Для проверки на четность можно воспользоваться предикатом `EVENP` или функцией нахождения остатка от деления `REM`.

Например,  $(-2\ 6\ s\ -1\ 4\ f\ 0\ z\ x\ r) \rightarrow (-2\ s\ 4\ 0\ x)$ .

18) Вычисляющую сумму элементов с нечетными номерами числового списка (нумерация элементов должна начинаться с 1). Для проверки на четность можно воспользоваться предикатом `EVENP` или функцией нахождения остатка от деления `REM`.

Например,  $(-2\ 3\ 2\ 5\ -6\ 5\ 2\ 1\ 3) \rightarrow -1$ .

19) Добавляющую элемент в конец числового списка, чтобы сумма элементов этого списка стала равна 100.

Например,  $(2\ 10\ 15\ 50\ 30) \rightarrow (2\ 10\ 15\ 50\ 30\ -7)$ .

20) Вычисляющую количество четных элементов в списке.

Например,  $(-2\ 6\ s\ -1\ 4\ f\ 0\ z\ x\ r) \rightarrow 4$ .

21) Формирующую список, состоящий из сумм первого и последнего, второго и предпоследнего элементов числового списка и т.д. Каждый элемент должен участвовать в сложении не более одного раза.

Например,  $(1\ -2\ -3\ 4\ 5\ 6\ -7\ 8\ 9) \rightarrow (10\ 6\ -10\ 10\ 5)$ .

22) Удаляющую элементы из первого списка с номерами из второго списка (второй список упорядочен по возрастанию, нумерация элементов должна начинаться с 1).

Например,  $L_1=(a\ s\ d\ f\ g\ h\ j\ k\ l)$ ,  $L_2=(1\ 4\ 5\ 8) \rightarrow (s\ d\ h\ j\ l)$ .

23) Формирующую подсписок из  $n$  элементов списка  $L$ , начиная с  $k$ -го элемента. (нумерация элементов должна начинаться с 1).

Например,  $L=(-2\ 6\ s\ -1\ 4\ f\ 0\ z\ x\ r)$ ,  $k=3$ ,  $n=4 \rightarrow (s\ -1\ 4\ f)$ .

24) Переставляющую элементы списка таким образом, чтобы одинаковые элементы оказались рядом. Сортировку не использовать!

Например,  $(1\ 5\ 2\ 1\ 4\ 3\ 1\ 2\ 4\ 5\ 4) \rightarrow (1\ 1\ 1\ 5\ 5\ 2\ 2\ 4\ 4\ 4\ 3)$ .

25) Преобразующую список  $L$  в новый список, элементы которого имеют вид:  $\langle \text{элемент списка } L \rangle \langle \text{кол-во вхождений этого элемента в список } L \rangle$ .

Например,  $L = (a\ b\ a\ a\ c\ b) \rightarrow ((a\ 3)\ (b\ 2)\ (c\ 1))$ .

Указание: Напишите вспомогательную функцию, которая подсчитывает количество вхождений элемента в список.

26) Вычисляющую количество атомов в списочной структуре (на всех уровнях).

Например,  $((a\ b)\ c\ (((d\ a\ v)))) \rightarrow 6$ .

27) Вычисляющую глубину списка.

Например,  $(((((1))))\ 2\ (3\ 4)) \rightarrow 4$ .

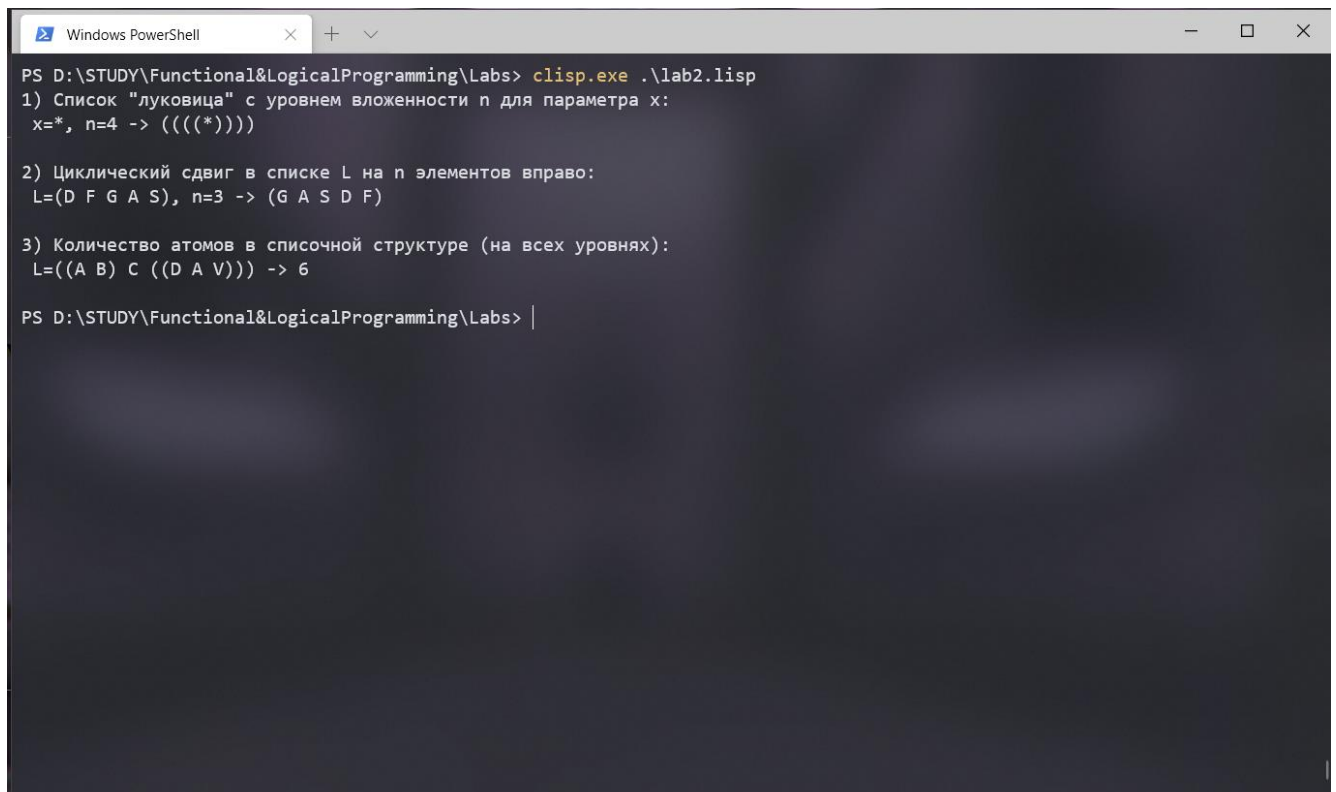
28) Преобразующую инфиксную запись выражения в прединфиксную и возвращающую значение выражения.

Например,  $((-8 + 10) * (12 / 3)) \rightarrow 8$ .

29) Преобразующую одноуровневый список во вложенный по следующему правилу:  $(a\ s\ d\ f\ g) \rightarrow (a\ (s\ (d\ (f\ (g))))))$ .

30) Преобразующую одноуровневый список во вложенный по следующему правилу:  $(a\ s\ d\ f\ g) \rightarrow (((((a)\ s)\ d)\ f)\ g)$ .

## 2. Примеры работы программы



```
PS D:\STUDY\Functional&LogicalProgramming\Labs> clisp.exe .\lab2.lisp
1) Список "луковица" с уровнем вложенности n для параметра x:
x=*, n=4 -> ((((*))))

2) Циклический сдвиг в списке L на n элементов вправо:
L=(D F G A S), n=3 -> (G A S D F)

3) Количество атомов в списочной структуре (на всех уровнях):
L=((A B) C ((D A V))) -> 6

PS D:\STUDY\Functional&LogicalProgramming\Labs> |
```

## Приложение Листинг

```
;; Каждая бригада должна выполнить 3 задачи (номера выбираются из таблицы). В теле
;; функции использование операторов set, let и setq не допускается! Все функции должны
;; быть рекурсивными, функционалы не использовать.
;; Бригада №6: 6, 16, 26
```

```
;; 6) Строящую список "луковица" с уровнем вложенности n для параметра x.
```

```
;; Например, n=4, x=* -> ((((*))))
```

```
(defun onion (x n)
```

```
  (if (eql n 0)
```

```
      x
```

```
      (list (onion x (- n 1))))
```

```
  )
```

```
)
```

```
(setq x "*")
```

```
(setq n 4)
```

```
(format t "1) Список \"луковица\" с уровнем вложенности n для параметра x:~% x=~A, n=~A -> ~A~%~%\" x n (onion x n))
```

```
;; 16) Осуществляющую циклический сдвиг в списке L на n элементов вправо.
```

```
;; Например, L = (a s d f g), n = 3 -> (d f g a s)
```

```
(defun cyclicShift (L n)
```

```
  (if (eql n 0)
```

```
      L
```

```
      (cyclicShift (append (last L) (butlast L)) (- n 1)))
```

```
  )
```

```
)
```

```
(setq L '(d f g a s))
```

```
(setq n 3)
```

```
(format t "2) Циклический сдвиг в списке L на n элементов вправо:~% L=~A, n=~A -> ~A~%~%\" L n (cyclicShift L n))
```

```
;; 26) Вычисляющую количество атомов в списочной структуре (на всех уровнях).
```

```
;; Например, ((a b) c ((d a v))) -> 6
```

```
(defun numAtoms (L)
```

```
  (if (not (null L))
```

```
      (+
```

```
        (if (consp (CAR L))
```

```
            (numAtoms (CAR L))
```

```
            1
```

```
        )
```

```
        (numAtoms (CDR L))
```

```
        )
```

```
        0
```

```
      )
```

```
)
```

```
(setq L '((a b) c ((d a v))))
```

```
(format t "3) Количество атомов в списочной структуре (на всех уровнях):~% L=~A -> ~A~%~%\" L (numAtoms L))
```