

## Лабораторная работа №2 (язык Clisp) Рекурсия

Каждая бригада должна выполнить 3 задачи: по одной из каждого раздела (номер задачи из каждого раздела совпадает с номером бригады). В бригаде может быть не более 2 человек.

В теле функции использование операторов цикла, set, let, функционалов **не допускается!** Все функции должны быть **рекурсивными**.

### I. 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,  $x=a$ ,  $L=(8\ *\ a\ 6\ a\ 1) \rightarrow (3\ 5)$ .
- 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,  $(-2\ 6\ s\ -1\ 4\ f\ 0) \rightarrow (6\ s\ 4\ f\ 0)$ .

### II. Определите функцию

- 1) Объединяющую 2 списка в один, чередуя элементы списков.  
Например,  $L1=(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8)$ ,  $L2=(a\ s\ d\ f) \rightarrow (1\ a\ 2\ s\ 3\ d\ 4\ f\ 5\ 6\ 7\ 8)$ .
- 2) Возвращающую список из пар соседних элементов, одинаково отстоящих от начала и конца списка. Например,  $(1\ a\ b\ c\ 2\ 3\ f) \rightarrow ((1\ f)\ (a\ 3)\ (b\ 2)\ (c))$
- 3) Определяющую, сколько раз заданное  $s$ -выражение входит в список.  
Например,  $x=(a)$ ,  $L=(1\ (a)\ x\ (a)\ 2\ a\ 1\ 2\ d) \rightarrow 2$ .
- 4) Преобразующую список в множество (для повторяющихся элементов должно оставаться последнее вхождение в список). Например,  $(a\ b\ a\ a\ c\ c) \rightarrow (b\ a\ c)$ .
- 5) Осуществляющую циклический сдвиг в списке  $L$  на  $n$  элементов влево.  
Например,  $L=(a\ s\ d\ f\ g)$ ,  $n=3 \rightarrow (f\ g\ a\ s\ d)$ .
- 6) Осуществляющую циклический сдвиг в списке  $L$  на  $n$  элементов вправо.

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

7) 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,  $(-2\ 6\ s\ -1\ 4\ f\ 0\ z\ x\ r) \rightarrow (-2\ s\ 4\ 0\ x)$ .

8) 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,  $(-2\ 3\ 2\ 5\ -6\ 5\ 2\ 1\ 3) \rightarrow -1$ .

9) Добавляющую элемент в конец числового списка, чтобы сумма элементов этого списка стала равна 100. Например,  $(2\ 10\ 15\ 50\ 30) \rightarrow (2\ 10\ 15\ 50\ 30\ -7)$ .

10) Вычисляющую количество четных элементов в списке. Для проверки на четность можно воспользоваться предикатом **EVENP** или функцией нахождения остатка от деления **REM**. Например,  $(-2\ 6\ s\ -1\ 4\ f\ 0\ z\ x\ r) \rightarrow 4$ .

### III. Определите функцию

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

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

Например,  $L1=(a\ s\ d\ f\ g\ h\ j\ k\ l)$ ,  $L2=(1\ 4\ 5\ 8) \rightarrow (s\ d\ h\ j\ l)$ .

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

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

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

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

5) 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.

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

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

7) Вычисляющую глубину списка. Например,  $(((((1))))\ 2\ (3\ 4)) \rightarrow 4$ .

8) Преобразующую инфиксную запись выражения в прединфиксную и возвращающую значение выражения. Например,  $((-8 + 10) * (12 / 3)) \rightarrow 8$ .

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

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