МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

дисциплина «Базы знаний и поддержка принятия решений в САПР »

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

по теме «Язык программирования ЛИСП»

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.

Принял: профессор

Мурашко И. А.

Гомель 2021

**Цель работы:** закрепление навыков работы со списками. Разработать программы для решения следующих задач:

1. Даны два списка вида (a1,a2,…) и (b1,b2,…). Получить список (a1,b1,a2,b2…). Если исходные списки разной длины, то остаток более длинного отбросить.

2. Даны натуральные m, nи некоторый список. Удалить из списка элементы с номерами с m-го по n-й.

3. Из произвольного списка удалить все элементы, не являющиеся числом

4. В сложном списке заменить все атомы-числа: положительные на слово “положительное” отрицательные на слово “отрицательное”, нулевые на “ноль”.

5. В произвольном списке удалить элементы с номерами **N** и **M** (без учета скобок)

**Выполнение:**

1. Даны два списка вида (a1,a2,…) и (b1,b2,…). Получить список (a1,b1,a2,b2…). Если исходные списки разной длины, то остаток более длинного отбросить.

(defun prog1(list1 list2)

(cond

((null list1) nil)

((null list2) nil)

(T (cons (car list1) (cons (car list2) (prog1 (cdr list1) (cdr list2)))))

)

)

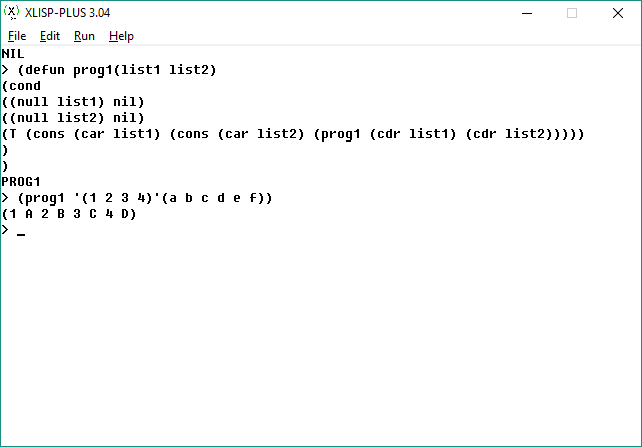


Рисунок 1 – Результат выполнения

2. Даны натуральные m, n и некоторый список. Удалить из списка элементы с номерами с m-го по n-й.

(defun z2 (lst m n)

(if (> m n)

(z2 m n)

(append (subseq lst 0 m) (subseq lst (+ n 1)))))

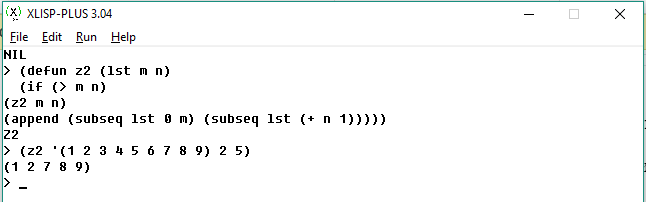


Рисунок 2 – Результат выполнения

3. Из произвольного списка удалить все элементы, не являющиеся числом

(defun z3 (lst)

(cond ((null lst) nil)

((symbolp (car lst)) (z3 (cdr lst)))

(t (cons (car lst) (z3 (cdr lst))))))

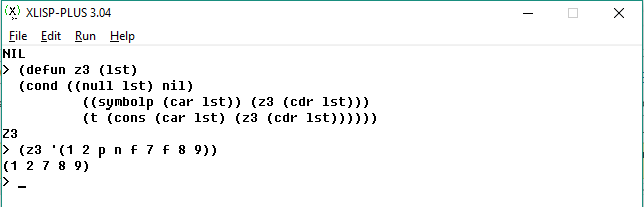


Рисунок 3 – Результат выполнения

4. В списке заменить все атомы-числа: положительные на слово

“положительное” отрицательные на слово “отрицательное”, нулевые на “ноль”.

(defun replace(list)

(cond

((null list) nil)

((listp (car list)) (replace list))

((> (car list) 0) (cons "Positive" (replace (cdr list))))

((< (car list) 0) (cons "Negative" (replace (cdr list))))

(T (cons "Zero" (replace (cdr list))))

)

)

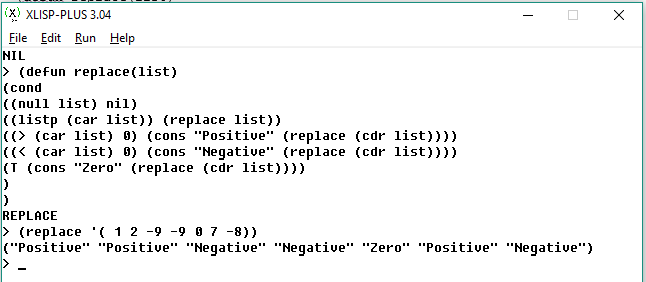


Рисунок 4 – Результат выполнения

5. В произвольном списке удалить элементы с номерами N и M.

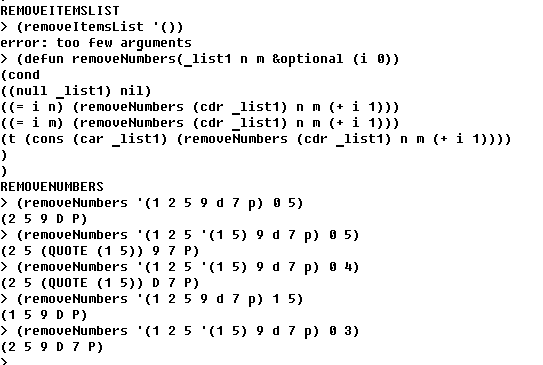


Рисунок 5 – Результат выполнения

**Задание 2**

1. Удалить первый и последний элементы сложного списка (без учета скобок).

2. Произвольный список вида (а1,а2,...,ак) разбить на два подсписка (а1,а3,а5...) и

(а2,а4,а6...).

3. Из произвольного списка удалить все отрицательные элементы.

4. Реверсировать произвольный список (включая подсписки).

5. Из произвольного списка удалить все латинские буквы, расположенные между

буквами **d** и **k**.

1. Удалить первый и последний элементы сложного списка (без учета скобок).

(defun z21 (lst)

(listfun (next lst))

)

(defun next (lst)

(cond

((listp (car lst))

(cons (next (car lst)) (cdr lst))

)

(

(cdr lst)

)

)

)

(defun listfun (lst)

(cond

((Null (cdr lst))

(cond

((listp (car lst))

(list (listfun (car lst)))

)

(

Nil

)

)

)

(

(cons (car lst) (listfun (cdr lst)))

)

)

)



Рисунок 6 – Результат выполнения

1. Произвольный список вида (а1,а2,...,ак) разбить на два подсписка (а

1,а3,а5...) и (а2,а4,а6...).

(defun lsplit (list)

(lsplit2 nil nil list)

)

(defun lsplit2 (list1 list2 list)

(cond

((null list) (values list1 list2))

(T (lsplit2 list2 (append list1 (list (car list))) (cdr list)) )

) )

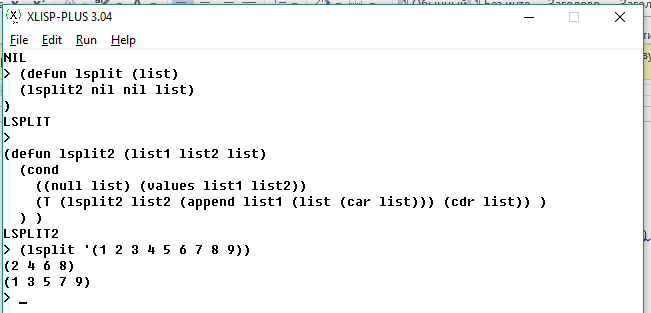


Рисунок 7 – Результат выполнения

1. Из произвольного списка удалить все отрицательные элементы.

(defun deln (list)

(cond

((null list) nil)

((listp (car list)) (cons (deln (car list)) (deln (cdr list))))

((>= (car list) 0) (cons (car list) (deln (cdr list))))

(T (deln (cdr list)))

)

)

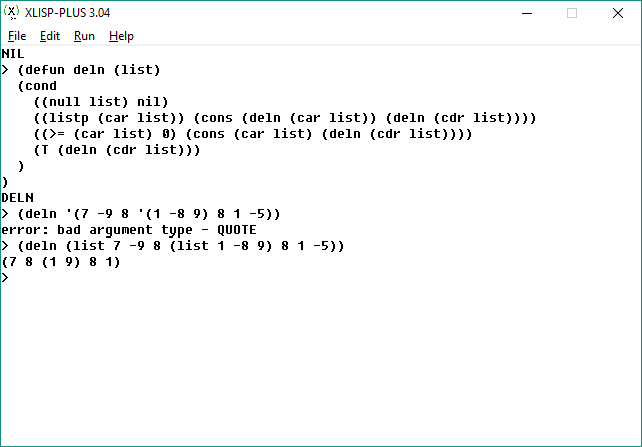


Рисунок 8 – Результат выполнения

1. Реверсировать произвольный список (включая подсписки).

(defun rlist (l)

(cond

((null l) nil)

((listp (car l)) (append (rlist (cdr l)) (list (rlist (car l)))))

(T (append (rlist (cdr l)) (list (car l))))

)

)

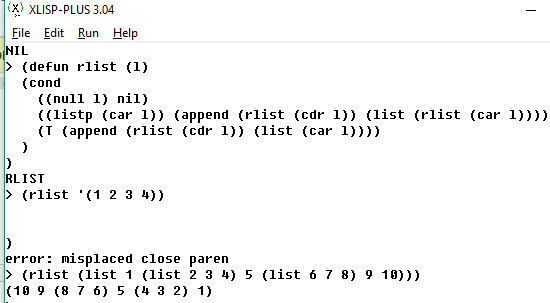


Рисунок 9 – Результат выполнения

1. Из произвольного списка удалить все латинские буквы, расположенные между буквами d и k.

(defun d(x)

(cond

((Null x)

Nil

)

((symbolp (car x))

(cond

((and (string< (car x) 'k) (string< 'd (car x)))

(d(cdr x))

)

(

(cons (car x) (d (cdr x)))

)

)

)

((listp (car x))

(cons (d (car x)) (d (cdr x)))

)

(

(cons (car x) (d (cdr x)))

)

)

)



Рисунок 10 – Результат выполнения

**Вывод:** в ходе выполнения лабораторной работы закреплены навыки работы со списками, разработаны программы для решения задач с простыми и сложными списками.