**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине «Базы знаний и поддержка принятия решений в системах автоматизированного проектирования»

на тему: «Числовые функции языка ЛИСП. Циклы и функционалы в ЛИСПе»

Выполнил: студент гр. ИТП-31

Трофимов Е.В.

Принял: преподаватель-стажёр

Васюкова В.О.

Гомель 2019

**Цель работы:** закрепление навыков работы с числовыми функциями.

**Задание 1.** Вычислить среднее арифметическое отрицательных чисел произвольного списка.

**Листинг программы:**

(defun task1 (mylist &optional (avg 0) (n 0))

(cond ((null mylist) (/ avg n))

((not (numberp (car mylist))) (task1 (cdr mylist) (+ (task1 (car mylist)) avg) (+ n 1)))

((< (car mylist) 0) (task1 (cdr mylist) (+ (car mylist) avg) (+ n 1)))

(t (task1 (cdr mylist) avg n))))

**Результат выполнения:**

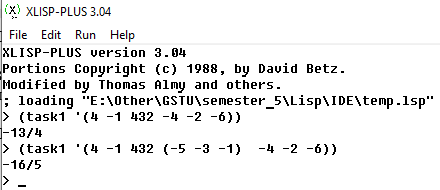


Рисунок 1 – Результат выполнения программы

**Задание 2.** Представить целое N в виде суммы квадратов двух целых чисел.

**Листинг программы:**

(defun quad (a b)

(+ (\* a a) (\* b b)))

(defun task2 (num &optional (a 0) (b 0))

(cond ((= (quad a b) num) (format t "~a^2+~a^2=~a" a b num))

((< b num) (task2 num a (+ b 1)))

((and (= b num) (< a num)) (task2 num (+ a 1) 0))

((and (= b num) (= a num)) "impossible") ))

(defun task2 (num &optional (a 0) (b 0))

(cond ((= (quad a b) num) (format t "~a^2+~a^2=~a" a b num))

((>= a num) "impossible")

((< b num) (task2 num a (+ b 1)))

((< a num) (task2 num (+ a 1) 0))

(t (format t "~a ~a" a b))))

**Результат выполнения:**

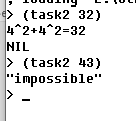


Рисунок 2 – Результат выполнения программы

**Задание 3.** Представить целое N в виде суммы квадратов 3-х натуральных чисел.

**Листинг программы:**

(defun quad (a b c)

(+ (\* a a) (\* b b) (\* c c)))

(defun task3 (num &optional (a 0) (b 0) (c 0))

(cond ((= (quad a b c) num) (format t "~a^2+~a^2+~a^2=~a" a b c num))

((and (>= a num) (>= b num) (>= c num)) "impossible")

((< c num) (task3 num a b (+ c 1)))

((and (= c num) (< b num)) (task3 num a (+ b 1) 0))

((= b num) (task3 num (+ a 1) 0 0))))

**Результат выполнения:**

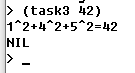


Рисунок 3 – Результат выполнения программы

**Задание 4.** Представить целое N в виде суммы квадратов 4-х натуральных чисел.

**Листинг программы:**

(defun quad (a b c d)

(+ (\* a a) (\* b b) (\* c c) (\* d d)))

(defun task4 (num &optional (a 0) (b 0) (c 0) (d 0))

(cond ((= (quad a b c d) num) (format t "~a^2+~a^2+~a^2+~a^2=~a" a b c d num))

((and (>= a num) (>= b num) (>= c num) (>= d num)) "impossible")

((< d num) (task4 num a b c (+ d 1)))

((and (< c num) (< b num)) (task4 num a b (+ c 1) 0))

((and (= c num) (< b num)) (task4 num a (+ b 1) 0 0))

((= b num) (task4 num (+ a 1) 0 0 0))))

**Результат выполнения:**

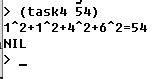


Рисунок 4 – Результат выполнения программы

**Задание 5.** Определить предикат, который проверяет, является ли произвольный список монотонной последовательностью чисел, или нет (без учета скобок).

**Листинг программы:**

(defun check-down (mylist &optional (last-el (car (cdr mylist))))

(cond ((null mylist) T)

((and (not (numberp (car mylist))) (check-down (cdr mylist))) )

((<= last-el (car mylist)) (check-down (cdr mylist) (car (cdr mylist)) ))

(NIL)))

(defun check-up (mylist &optional (last-el (car (cdr mylist))) (check T))

(cond ((null mylist) T)

((and (not (numberp (car mylist))) (check-up (car mylist))) (check-up (cdr mylist) (car (cdr mylist)) NIL))

((>= last-el (car mylist)) (check-up (cdr mylist) (car (cdr mylist)) NIL))

(check (check-down mylist))

(NIL)))

(defun task5 (mylist)

(cond ((null mylist) NIL)

(t (check-up mylist))))

**Результат выполнения:**

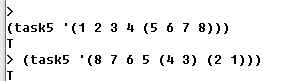


Рисунок 5 – Результат выполнения программы

**Задание 6.** Найти максимальную глубину вложенности произвольного списка.

**Листинг программы:**

(defun task6 (mylist &optional (lvl 0) (maxLvl 0))

(cond ((null mylist) lvl)

((not (numberp (car mylist))) (task6 (cdr mylist)

(if (> (task6 (car mylist) (+ lvl 1)) lvl) (task6 (car mylist) (+ lvl 1)) lvl)))

(t (task6 (cdr mylist) lvl))))

**Результат выполнения:**

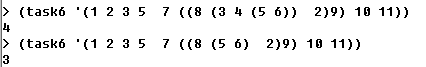


Рисунок 6 – Результат выполнения программы

**Задание 7.** Используя управляющую конструкцию DO, вычислить среднее арифметическое чисел 0.1+0.25+…+15.1.

**Листинг программы:**

(defun task7 ()

(do ((x 0.1 (+ x 0.15)) (n 0 (+ n 1)) )

((>= x 15.1) (/ x n)) ))

**Результат выполнения:**



Рисунок 7 – Результат выполнения программы

**Задание 8.** Используя управляющую конструкцию DOlist, вычислить среднее арифметическое и дисперсию чисел из простого списка

**Листинг программы:**

(defun task8 (mylist &optional (s 0) (d 0))

(progn

(dolist (i mylist)

(setq s (+ i s)))

(setq s (/ s (length mylist)))

(dolist (i mylist)

(setq d (+ (\* (- i s) (- i s)) d)))

(setq d (/ d (length mylist)))

(print s)

(print d)))

**Результат выполнения:**

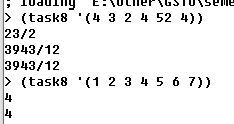


Рисунок 8 – Результат выполнения программы

**Задание 9.** Даны два списка вида (*a1,a2,…*) и (*b1,b2,…*). Используя отображающий функционал *MAPx*, получить список вида (*a1,b1,a2,b2…*). Если исходные списки разной длины, то остаток более длинного списка дописать в конец без изменения.

**Листинг программы:**

(defun con (list1 list2)

(cond ((and (null list1) (null list2)) NIL)

((not (null list1)) (cons (car list1) (con (cdr list1) list2)))

(t (cons (car list2) (con nil (cdr list2))))

))

(defun sub-list (mylist n &optional (i 0))

(cond ((null mylist) NIL)

((< i n) (sub-list (cdr mylist) n (+ i 1)))

(t (cons (car mylist) (sub-list (cdr mylist) n (+ i 1))) ) ))

(defun task9 (list1 list2 &optional (result NIL))

(cond ((= (length list1) (length list2)) (mapcan 'list list1 list2))

((> (length list1) (length list2))

(con (mapcan 'list list1 list2) (sub-list list1 (length list2))))

(t (con (mapcan 'list list1 list2) (sub-list list2 (length list1)))) ))

**Результат выполнения:**

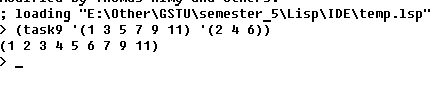


Рисунок 9 – Результат выполнения программы

**Задание 10.** Разработать функцию преобразования и вычисления математических выражений в префиксную форму.

**Листинг программы:**

(defun do-operation(a sign b)

(cond ((string= sign "+") (+ a b))

((string= sign "-") (- a b))

((string= sign "\*") (\* a b))

((string= sign "/") (/ a b))))

(defun task10(mylist)

(cond((null (cdr mylist)) (car mylist))

(t (do-operation (car mylist) (car (cdr mylist)) (task10 (cdr (cdr mylist)))))))

**Результат выполнения:**

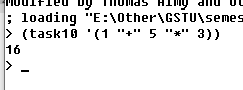


Рисунок 10 – Результат выполнения программы

**Вывод:** закрепили навыки работы с числовыми функциями и функциями работы со списками, циклами и предикатами типов данных.