Министерство науки и высшего образования Российской Федерации   
Федеральное государственное бюджетное образовательное   
учреждение высшего образования   
«Алтайский государственный технический   
университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Андреева А.Ю.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Отчёт

по лабораторной работе № 1

«ИСПОЛЬЗОВАНИЕ ВСТРОЕННЫХ ФУНКЦИЙ ЛИСПА»

по дисциплине «Функциональное и логическое программирование»

Студент группы ПИ-92 Шульпов В.М.

Преподаватель доцент, к.ф.-м.н. Андреева А.Ю.

Барнаул 2022

**Вариант 29**

Необходимо выполнить по одному заданию из каждого упражнения

(вариант взять у преподавателя).

Задания выполняются в среде сLisp (или другой). Результаты привести в

отчете. В случае ошибок интерпретации – привести пояснения.

Быть готовым пояснить любой вопрос по тексту кода.

Подсчёт варианта 29+2 = 31, 31%30=1

**Упражнение 1**

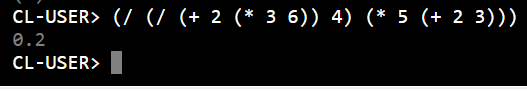
Записать следующие выражения в виде s-выражений Лиспа:

**№1 (1/30)**

(2+3\*6)/4/(5\*(2+3))

= 0.2

(/ (/ (+ 2 (\* 3 6)) 4) (\* 5 (+ 2 3)))



**Упражнение 2**

Определить, как будут выполнены интерпретатором Лиспа следующие

последовательности s-выражений, т.е. что появится на выходе и какие значения

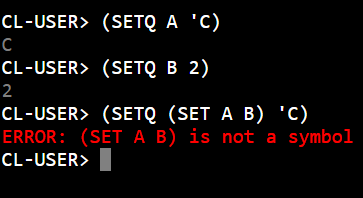
получат символьные атомы в результате присваиваний

**№3 (3/15)**

(SETQ A ‘C) -> C (в A запишется ссылка на С)

(SETQ B 2) -> 2 (в B запишется число 2)

(SETQ (SET A B) 'C) -> ошибка, т.к. (SET A B) вернёт число 2, оно станет первым аргументом функции SETQ, который должен быть символьным атомом, а не числом



**Упражнение 3**

Записать последовательность функций CAR и CDR, выделяющую из

приведенных списков атом A.

**№5 (5/30)**

(1 (2) 3 (A))

(CAR (CADDDR ‘(1 (2) 3 (A)))) или (CAR (CAR (CDR (CDR (CDR ‘(1 (2) 3 (A))))))))

****



**Упражнение 4**

Каков результат интерпретации следующих выражений?

**(7/23)**

(CAR (SET ‘A ‘(B C D))) -> B

По ссылке А запишется список (B C D), затем из этого списка будет возвращена «голова», т.е. первый элемент списка B



**Упражнение 5**

В данном упражнении требуется получить списки справа от стрелки с

помощью суперпозиции функций LIST, APPEND, CONS, CAR, CDR.

Аргументами функций могут быть лишь s-выражения, указанные слева от

стрелки.

Пример: (A B); C; D => (A B (C D))

Решение: (APPEND '(A B) (LIST (LIST ‘C ‘D)))

**(9/18)**

(A); (B); (C D) => (A (B C) D)

(CAR '(A)) -> A

(APPEND '(B) (LIST (CAR '(C D)))) -> (B C)

(CADR '(C D)) -> D

А потом все это объединяем с помощью LIST



**Упражнение 6**

Какое из выражений дает значение T, а какое NIL?

**(11/30)**

(ATOM (SETQ A 'B))

Вернёт T (истину), т.к. функция (SETQ A ‘B) вернёт B (это символьный атом), а функция ATOM проверяет, является ли ее аргумент атомом.

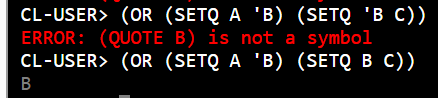


**Упражнение 7**

Чему равны значения следующих выражений?

**(13/30)**

(OR (SETQ A ‘B) (SETQ ‘B C))

м

(SETQ ‘B C) вызовет ошибку, т.к. функция SETQ не требует апострофа у первого аргумента (SETQ A B) ⬄ (SET ‘A B)

Если уберем апостроф, то вернется первый истинный аргумент, т.е. B

**Упражнение 8**

Записать определения следующих функций, используя функции проверки

условия IF и COND (т.е. 2 варианта!)

**(15/15)**

Аргументом функции является список из трех чисел. Функция должна

вернуть список из тех же чисел, расположенных в порядке возрастания.

**Через IF:**

//SORT1-2 принимает список из 3 элементов и сортирует в нём первые 2 по возрастанию

(возвращает измененный список)

CL-USER> (DEFUN SORT1-2 (x) (IF (> (CAR x) (CADR x)) (LIST (CADR x) (CAR x) (CADDR x)) x))

SORT1-2

CL-USER> (SORT1-2 '(3 2 1))

(2 3 1)

//SORT2-3 принимает список из 3 элементов и сортирует в нём последние 2 по возрастанию (возвращает измененный список)

CL-USER> (DEFUN SORT2-3 (x) (IF (> (CADR x) (CADDR x)) (LIST (CAR x) (CADDR x) (CADR x)) x))

SORT2-3

CL-USER> (SORT2-3 '(3 2 1))

(3 1 2)

//SORT3 принимает список из 3 элементов и сортирует его по возрастанию

CL-USER> (DEFUN SORT3 (x) ( SORT2-3 ( SORT1-2 (SORT2-3 (SORT1-2 x) )) ) )

SORT3

CL-USER> (SORT3 '(3 2 1))

(1 2 3)

CL-USER> (SORT3 '(1 2 3))

(1 2 3)

CL-USER> (SORT3 '(2 1 3))

(1 2 3)

CL-USER> (SORT3 '(2 3 1))

(1 2 3)

CL-USER>

**Через COND:**

CL-USER> (DEFUN SORT1-2 (x) (COND ((> (CAR x) (CADR x)) (LIST (CADR x) (CAR x) (CADDR x))) (T x)))

SORT1-2

CL-USER> (SORT1-2 '(3 2 1))

(2 3 1)

CL-USER> (SORT1-2 '(1 2 3))

(1 2 3)

CL-USER> (DEFUN SORT2-3 (x) (COND ((> (CADR x) (CADDR x)) (LIST (CAR x) (CADDR x) (CADR x))) (T x)))

SORT2-3

CL-USER> (SORT2-3 '(3 2 1))

(3 1 2)

CL-USER> (SORT2-3 '(1 2 3))

(1 2 3)

CL-USER> (SORT3 '(3 2 1))

(1 2 3)

CL-USER>