|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 10 |

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | Н.А. Гарасев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Б.Толпинская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

**Задание 8. Rec-add**

**(defun** rec-add **(**lst**)**

**(cond** **((null** lst**)** 0**)**

**(**T **(**r-add lst 0**))))**

**(defun** r-add **(**lst res**)**

**(cond** **((null** lst**)** res**)**

**(**T **(**r-add **(cdr** lst**)** **(+** res **(car** lst**))))))**

\* (rec-add `(1 2 3 4 5))

15

\* (rec-add `())

0

**Задание 9. Rec-nth**

**(defun** rec\_nth **(**lst n**)**

**(cond** **((null** lst**)** Nil**)**

**((=** n 0**)** **(car** lst**))**

**(**T **(**my\_nth **(cdr** lst**)** **(-** n 1**)))))**

\* (rec\_nth `(1 2 3 4 5 6) 2)

3

\* (rec\_nth `(1 2 3 4 5 6) 8)

NIL

**Задание 10. Alloddr**

**(defun** alloddr **(**lst**)**

**(cond** **((null** lst**)** T**)**

**((oddp** **(car** lst**))** **(**alloddr **(cdr** lst**)))))**

\* (alloddr `(1 2 3 4 5 6))

NIL

\* (alloddr `(2 4 6 8))

NIL

\* (alloddr `(1 3 5 7 9))

T

**Задание 11. Хвостовая рекурсия. Last.**

**(defun** hvost **(**lst**)**

**(cond** **((null** **(cdr** lst**))** **(car** lst**))**

**(**T **(**hvost **(cdr** lst**)))))**

\* (hvost `(1 2 3 4 5))

5

\* (hvost `())

NIL

**Задание 12. Сумму с 0 по n. Сумму с n до последнего>= 0. Сумма с n по m с шагом d.**

**(defun** sum1 **(**lst n**)**

**(**sum1-inner lst n 0**))**

**(defun** sum1-inner **(**lst n res**)**

**(cond** **((or** **(null** lst**)** **(<=** n 0**))** res**)**

**(**T **(**sum1-inner **(cdr** lst**)** **(-** n 1**)** **(+** res **(car** lst**))))))**

**(defun** sum2 **(**lst n**)**

**(**sum2-inner lst n 0 0**))**

**(defun** sum2-inner **(**lst n res tmp**)**

**(cond** **((null** lst**)** res**)**

**(**T **(**sum2-inner **(cdr** lst**)**

**(-** n 1**)**

**(cond** **((>** n 1**)** 0**)**

**((<** **(car** lst**)** 0**)** res**)**

**((>** **(car** lst**)** 0**)** **(+** res **(+** **(car** lst**)** tmp**)))**

**(**T **(+** res **(car** lst**))))**

**(cond** **((>** **(car** lst**)** 0**)** 0**)**

**(**T **(+** tmp **(car** lst**))))))))**

**(defun** sum3 **(**lst n m d**)**

**(**sum3-inner lst n m **(-** d 1**)** 0 0**))**

**(defun** sum3-inner **(**lst n m d res tmp**)**

**(cond** **((or** **(null** lst**)** **(<** m 0**))** res**)**

**(**T **(**sum3-inner **(cdr** lst**)**

**(-** n 1**)**

**(-** m 1**)**

d

**(cond** **((and** **(<=** n 0**)** **(=** tmp 0**))** **(+** res **(car** lst**)))**

**(**T res**))**

**(cond** **((or** **(=** tmp d**)** **(>** n 0**))** 0**)**

**((<=** n 0**)** **(+** 1 tmp**))**

**(**T 0**))))))**

\* (sum1 `(1 2 3 4 5 6) 4)

10

\* (sum2 `(1 2 3 4 5 6 -1) 4)

15

\* (sum2 `(1 2 3 4 5 6 -1 2) 4)

16

\* (sum3 `(1 2 3 4 5 6 7) 2 5 2)

8

\* (sum3 `(1 2 3 4 5 6 7) 2 5 1)

18

**Задание 13. Последнее нечетное число.**

**(defun** last-odd **(**lst**)**

**(let** **((**x **(**last-odd-inner **(cdr** lst**)** **(car** lst**))))** **(and** **(oddp** x**)** x**)))**

**(defun** last-odd-inner **(**lst od**)**

**(cond** **((null** lst**)** od**)**

**((oddp** **(car** lst**))** **(**last-odd-inner **(cdr** lst**)** **(car** lst**)))**

**(**T **(**last-odd-inner **(cdr** lst**)** od**))))**

\* (last-odd `(1 2 3 4 5 6 7 8 9))

9

\* (last-odd `(1 2 3 4 5 6 7 8))

7

**Задание 14. Cons-дополняемую рекурсия с одним тестом завершения. Список квадратов чисел.**

**(defun** cons-squad **(**lst**)**

**(cond** **((null** lst**)** Nil**)**

**(**T **(cons** **(\*** **(car** lst**)** **(car** lst**))** **(**cons-squad **(cdr** lst**))))))**

\* (cons-squad `(1 2 3 4 5 6 7 8 9))

(1 4 9 16 25 36 49 64 81)

**Задание 15. Функции выбора четных и нечетных, а также суммы четных и нечетных.**

**(defun** select-odd **(**lst**)**

**(**reduce **(lambda** **(**res a**)**

**(cond** **((and** **(numberp** a**)** **(oddp** a**))** **(append** res **(list** a**)))**

**(**T res**)))** lst **:initial-value** Nil**))**

**(defun** select-even **(**lst**)**

**(**reduce **(lambda** **(**res a**)**

**(cond** **((and** **(numberp** a**)** **(evenp** a**))** **(append** res **(list** a**)))**

**(**T res**)))** lst **:initial-value** Nil**))**

**(defun** sum-all-odd **(**lst**)**

**(**reduce **(lambda** **(**res a**)**

**(cond** **((and** **(numberp** a**)** **(oddp** a**))** **(+** res a**))**

**(**T res**)))** lst **:initial-value** 0**))**

**(defun** sum-all-even **(**lst**)**

**(**reduce **(lambda** **(**res a**)**

**(cond** **((and** **(numberp** a**)** **(evenp** a**))** **(+** res a**))**

**(**T res**)))** lst **:initial-value** 0**))**

\* (select-odd `(1 2 3 4 5 6 7 8 9))

(1 3 5 7 9)

\* (select-even `(1 2 3 4 5 6 7 8 9))

(2 4 6 8)

\* (sum-all-even `(1 2 3 4 5 6 7 8 9))

20

\* (sum-all-odd `(1 2 3 4 5 6 7 8 9))

25

**Дополнительное задание. Создать и обработать структурированный список. Изменить з/п в зависимости от заданного условия и подсчитать суммарную зарплату. Использовать композиции функций.**

**(setf** my\_struct **`((**Nikita 1000 20 lisp-developer**)** **(**Dima 900 19 lisp-tester**)** **(**Andrey 666 21 lisp-hater**)))**

**(defun** sum-salary **(**lst**)** **(**sum-salary-inner lst 0**))**

**(defun** sum-salary-inner **(**lst res**)**

**(cond** **((null** lst**)** res**)**

**(**T **(**sum-salary-inner **(cdr** lst**)** **(+** res **(car** **(cdr** **(car** lst**))))))))**

**(defun** change-salary **(**lst new n**)**

**(**reduce **(lambda** **(**res tmp**)**

**(cond** **((eq** **(**my\_len res**)** n**)** **(append** res **(list** **(list** **(car** tmp**)** new **(cadr** tmp**)(caddr** tmp**)))))**

**(**T **(append** res **(list** tmp**)))))**

lst **:initial-value** Nil**))**

\* (change-salary my\_struct 101 0)

((NIKITA 101 1000 20) (DIMA 900 19 LISP-TESTER) (ANDREY 666 21 LISP-HATER))

\* (sum-salary my\_struct)

1667