



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 «Программное обеспечение ЭВМ и информационные технологии»

Рубежный контроль № 2

Студент Лучина Е.Д

Группа ИУ7-61Б

Вариант 2

Преподаватель Толпинская Н.Б.

Москва.

2020 г.

Даны:

- смешанный структурированный список (т.е. любой элемент списка может быть: символ, число или список),
- множество, в виде смешанного одноуровневого (т.е. не структурированного) списка.

Задача:

На всех уровнях заданного списка найти количество элементов, принадлежащих заданному множеству. Заменить K-ый элемент верхнего уровня исходного списка на найденное значение или дописать это значение в конец исходного списка.

Функция обнаружения принадлежности.

```
(defun is_in (element m_set)
  (cond
    ((null m_set) Nil)
    ((eq element (car m_set)) T)
    (t (is_in element (cdr m_set))))
  )
)
```

Получает в качестве аргументов элемент (неструктурированный) и множество (смешанное одноуровневое). Возвращает nil если элемент не найден (включая случай пустого множества), T - если элемент найден. Работает рекурсивно. При нахождении вхождения заканчивает свою работу.

Функция подсчета элементов, принадлежащих заданному множеству.

```
(defun m_count (m_set m_list)
  (cond
    ((null m_list) 0)
    ((listp (car m_list))
     (+ (m_count m_set (car m_list))
        (m_count m_set (cdr m_list))
        ))
    ((is_in (car m_list) m_set)
     (+ 1 (m_count m_set (cdr m_list)))))
    (t (m_count m_set (cdr m_list)))
  )
)
```

Реализована с помощью дополняемой рекурсии. В качестве параметров принимает множество и список (описанные в задании). В случае пустого

списка возвращает 0. Если элемент тоже является списком, спускается на уровень ниже, возвращает сумму вхождений внутри этого элемента и внутри хвоста. Если элемент не является списком, то происходит проверка на принадлежность. Если элемент принадлежит множеству, то результат увеличивается на один.

Функция вставки элемента `elem` по индексу `k` в список `m_list`.

```
(defun m_insert (m_list k elem)
  (cond
    ((null m_list) (cons elem Nil))
    ((eq k 0) (cons elem m_list))
    (t (cons (car m_list) (m_insert (cdr m_list) (- k 1) elem)))
  )
)
```

Реализовано рекурсивно. Если `k` больше длины списка, элемент будет вставлен в конец.

Основная функция

```
(defun func (m_set m_list k)
  (m_insert m_list k (m_count m_set m_list))
)
```

Примеры работы

```
(func '(1 a 3 4) '(2 (3 5 h) (5 (5 (8) a) 3 1)) 0)
```

```
>> (4 2 (3 5 h) (5 (5 (8) A) 3 1))
```

```
(func '(1 a 3 4) '(2 (3 5 h) (5 (5 (8) a) 3 1)) 2)
```

```
>> (2 (3 5 h) 4 (5 (5 (8) A) 3 1))
```

```
(func '(1 a 3 4) '(2 (3 5 h) (5 (5 (8) a) 3 1)) 40)
```

```
>> (2 (3 5 h) (5 (5 (8) A) 3 1) 4)
```

```
(func '() '(2 (3 5 h) (5 (5 (8) a) 3 1)) 0)
```

```
>> (0 2 (3 5 h) (5 (5 (8) A) 3 1))
```

```
(func '(1 a 3 4) '() 0)
```

```
>> (0)
```

```
(func2 '() '() 0)
```

```
>> (0)
```

*Отсортировать заданное множество по возрастанию (для max-х баллов).

```
(defun sorted_insert (m_set elem)
  (cond
    ((null m_set) (cons elem Nil))
    ((< elem (car m_set)) (cons elem m_set))
    (t (cons (car m_set) (sorted_insert (cdr m_set) elem)))
  )
)
```

```
(defun m_sort (m_set res)
  (cond
    ((null m_set) res)
    (t (m_sort (cdr m_set) (sorted_insert res (car m_set))))
  )
)
```

```
(defun my_sort (m_set)
  (m_sort m_set Nil)
)
```