



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»

## Отчет по лабораторным работам №17 - 18 по дисциплине «Функциональное и логическое программирование»

Тема Обработка, формирование и модификация списков на Prolog

Студент Варламова Е.А.

Группа ИУ7-61Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Толпинская Н.Б., Строганов Ю. В.

# Лабораторная работа №17

## Постановка задачи

**Задание:** используя хвостовую рекурсию, разработать эффективную программу (комментируя назначение аргументов), позволяющую:

1. Найти длину списка (по верхнему уровню);
2. Найти сумму элементов числового списка;
3. Найти сумму элементов числового списка, стоящих на нечетных позициях исходного списка (нумерация от 0);

## Решение

```
1 domains
2   list = integer*
3
4 predicates
5   rec_length(integer, integer, list)
6   length(integer, list)
7
8   rec_sum(integer, integer, list)
9   sum(integer, list)
10
11  rec_oddsum(integer, integer, list)
12  oddsum(integer, list)
13
14 clauses
15  rec_length(Res, Len, [_ | Tail]) :- NewLen = Len + 1, rec_length(Res,
16    NewLen, Tail).
17  rec_length(Res, Len, []) :- Res = Len, !.
18  length(Res, List) :- rec_length(Res, 0, List).
19
20  rec_sum(Res, Sum, [Head | Tail]) :- NewSum = Sum + Head, rec_sum(Res,
21    NewSum, Tail).
22  rec_sum(Res, Sum, []) :- Res = Sum, !.
23  sum(Res, List) :- rec_sum(Res, 0, List).
```

```
23 rec_oddsum(Res, Sum, [_ , Head | Tail]) :- NewSum = Sum + Head, rec_oddsum(  
    Res, NewSum, Tail).  
24 rec_oddsum(Res, Sum, []) :- Res = Sum, !.  
25 oddsum(Res, List) :- rec_oddsum(Res, 0, List).  
26  
27 goal  
28   %length(Res, [1, 2, 3, 4]).  
29   %sum(Res, [1, 2, 3, 4]).  
30   %oddsum(Res, [1, 2, 3, 4]).
```

## Вопрос

Вопрос:  $\text{length}(\text{Res}, [1, 2])$

No шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	$\text{length}(\text{Res}, [1, 2])$	$\text{rec\_length}(\text{Res}, \text{Len}, [_   \text{Tail}])$ и $\text{length}(\text{Res}, [1, 2])$ <b>Результат:</b> неудача, термы не унифицируемы	Прямой ход. Переход к следующему предложению.
2	$\text{length}(\text{Res}, [1, 2])$	$\text{rec\_length}(\text{Res}, \text{Len}, [])$ и $\text{length}(\text{Res}, [1, 2, 3, 4])$ <b>Результат:</b> неудача, термы не унифицируемы	Прямой ход. Переход к следующему предложению.
3	$\text{length}(\text{Res}, [1, 2])$	$\text{length}(\text{Res}, [1, 2])$ и $\text{length}(\text{Res}, \text{List})$ <b>Результат:</b> успех + подстановка $\{\text{List} = [1, 2], \text{Res} = \text{Res}\}$	Заголовок правила заменяется его телом с учетом подстановки
4	$\text{rec\_length}(\text{Res}, 0, [1, 2]), !$	$\text{rec\_length}(\text{Res}, \text{Len}, [_   \text{Tail}])$ и $\text{rec\_length}(\text{Res}, 0, [1, 2], !)$ <b>Результат:</b> успех + подстановка $\{\text{Len} = 0, \text{Res} = \text{Res}, \text{Tail} = [2]\}$	Заголовок правила заменяется его телом с учетом подстановки
5	$\text{NewLen} = 0 + 1, \text{rec\_length}(\text{Res}, \text{NewLen}, [2]), !$	$\text{NewLen} = 0 + 1$ <b>Результат:</b> успех	Переход к следующему терму
6	$\text{rec\_length}(\text{Res}, 1, [2]), !$	$\text{rec\_length}(\text{Res}, 1, [2])$ и $\text{rec\_length}(\text{Res}, \text{Len}, [_   \text{Tail}])$ <b>Результат:</b> успех + подстановка $\{\text{Len} = 1, \text{Res} = \text{Res}, \text{Tail} = []\}$	Заголовок правила заменяется его телом с учетом подстановки:
7	$\text{NewLen} = 1 + 1, \text{rec\_length}(\text{Res}, \text{NewLen}, []), !$	$\text{NewLen} = 1 + 1$ <b>Результат:</b> успех	Переход к следующему терму
8	$\text{rec\_length}(\text{Res}, 2, []), !$	$\text{rec\_length}(\text{Res}, \text{Len}, [_   \text{Tail}])$ и $\text{rec\_length}(\text{Res}, 2, [])$ <b>Результат:</b> неудача, термы не унифицируемы	Прямой ход. Переход к следующему предложению.
9	$\text{rec\_length}(\text{Res}, 2, []), !$	$\text{rec\_length}(\text{Res}, 2, [])$ и $\text{rec\_length}(\text{Res}, \text{Len}, [])$ <b>Результат:</b> успех + подстановка $\{\text{Len} = 2, \text{Res} = \text{Res}\}$	Заголовок правила заменяется его телом с учетом подстановки
10	$\text{Res} = 2, !, !$	$\text{Res} = 2$ <b>Результат:</b> успех	Переход к следующему терму
11	$!, !$	$!$ <b>Результат:</b> успех	Переход к следующему терму

Рис. 1: Таблица к вопросу

12	!	! <b>Результат: успех</b>	Переход к следующему терму
13			Найдено решение Res=2 Отсечение => система больше не будет искать решение в процедуре rec_length

Рис. 2: Таблица к вопросу

# Лабораторная работа №18

## Постановка задачи

**Задание:** используя хвостовую рекурсию, разработать, комментируя аргументы, эффективную программу, позволяющую:

1. Сформировать список из элементов числового списка, больших заданного значения;
2. Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);
3. Удалить заданный элемент из списка (один или все вхождения);
4. Преобразовать список в множество (можно использовать ранее разработанные процедуры).

## Решение

```
1 domains
2   list = integer*
3
4 predicates
5   bigger(list, integer, list)
6   odd_list(list, list)
7   set(list, list)
8   rm_all(list, integer, list)
9   rm_one(list, integer, list)
10
11 clauses
12   odd_list([_, Head | Tail], [Head | ResTail]) :- odd_list(Tail, ResTail).
13   odd_list([], []) :- !.
14
15   bigger([Head | Tail], N, [Head | ResTail]) :- Head > N, !, bigger(Tail, N,
16                               ResTail).
17   bigger([_ | Tail], N, Result) :- bigger(Tail, N, Result).
18   bigger([], _, []).
19
20   rm_one([Head | Tail], N, Tail) :- Head = N, !.
21   rm_one([Head | Tail], N, [Head | ResList]) :- !, rm_one(Tail, N, ResList).
22   rm_one([], _, []).
```

```

22
23   rm_all([Head | Tail], N, [Head | ResList]) :- Head <> N, !, rm_all(Tail, N
    , ResList).
24   rm_all([_ | Tail], N, ResList) :- rm_all(Tail, N, ResList), !.
25   rm_all([], _, []).
26
27   set([Head | Tail], [Head | Result]) :- rm_all(Tail, Head, Nt), !, set(Nt,
    Result).
28   set([], []).
29
30 goal
31   %bigger([1, 7, 3, 4, 5, 6, 2], 3, Result).
32   %odd_list([1, 2, 3, 4, 5, 6, 7, 8], Result).
33
34   %rm_one([1, 2, 3, 1, 2, 3, 1, 2, 3], 3, Result).
35   %rm_all([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
36
37   %set([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).

```

## Вопрос

Вопрос: `odd_list([1, 2, 3, 4], Result)`

No шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	<code>odd_list([1, 2, 3, 4], Result)</code>	<code>odd_list([_, Head   Tail], [Head   ResTail])</code> и <code>odd_list([1, 2, 3, 4], Result)</code> <b>Результат:</b> успех + подстановка $\{\_ = 1, \text{Head} = 2, \text{Tail} = [3, 4], \text{Result} = [2   \text{Restail}]\}$	Заголовок правила заменяется его телом с учетом подстановки
2	<code>odd_list([3, 4], ResTail)</code>	<code>odd_list([_, Head   Tail], [Head   ResTail])</code> и <code>odd_list([3, 4], Result)</code> <b>Результат:</b> успех + подстановка $\{\_ = 3, \text{Head} = 4, \text{Tail} = [], \text{Result} = [4   \text{Restail}]\}$	Заголовок правила заменяется его телом с учетом подстановки
3	<code>odd_list([], ResTail)</code>	<code>odd_list([_, Head   Tail], [Head   ResTail])</code> и <code>odd_list([3, 4], Result)</code> <b>Результат:</b> неудача, термы не унифицируемы	Переход к следующему терму
4	<code>odd_list([], ResTail)</code>	<code>odd_list([], [])</code> и <code>odd_list([], ResTail)</code> <b>Результат:</b> успех + подстановка $\{\text{Restail} = []\}$	Заголовок правила заменяется его телом с учетом подстановки
5	!	успех	Переход к следующему терму
6			Решение найдено, возврат результата <code>Restail = []</code>

Рис. 3: Таблица к вопросу