



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 лабораторной работе №17 по курсу  
«Функциональное и логическое  
программирование»**

Тема Обработка списков на Prolog

Студент Климов И.С.

Группа ИУ7-62Б

Оценка (баллы) \_\_\_\_\_

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

Москва — 2022 г.

## Задание 1

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).

Убедиться в правильности результатов

## Решение

### domains

```
list = integer*.
```

### predicates

```
list_length(list, unsigned).
```

```
list_length_help(list, unsigned, unsigned).
```

```
list_sum(list, integer).
```

```
list_sum_help(list, integer, integer).
```

```
sum_odd(unsigned, integer, integer, integer).
```

```
list_sum_odd(list, integer).
```

```
list_sum_odd_help(list, integer, unsigned, integer).
```

### clauses

```
list_length_help([], Sum, Sum) :- !.
```

```
list_length_help([_|T], Length, Sum) :-
```

```
    UpdSum = Sum + 1,
```

```
    list_length_help(T, Length, UpdSum).
```

```
list_length(List, Length) :- list_length_help(List, Length, 0).
```

```
list_sum_help([], Sum, Sum) :- !.
```

```
list_sum_help([H|T], ResultSum, Sum) :-
```

```
    UpdSum = Sum + H,
```

```
    list_sum_help(T, ResultSum, UpdSum).
```

```
list_sum(List, Sum) :- list_sum_help(List, Sum, 0).
```

```

sum_odd(Position, Number, Sum, NewSum) :-
    Position mod 2 = 1,
    NewSum = Sum + Number,
    !.
sum_odd(_, _, Sum, Sum) :- !.

list_sum_odd_help([], Sum, _, Sum) :- !.
list_sum_odd_help([H|T], ResultSum, Position, Sum) :-
    sum_odd(Position, H, Sum, UpdSum),
    IncPosition = Position + 1,
    list_sum_odd_help(T, ResultSum, IncPosition, UpdSum).
list_sum_odd(List, Sum) :- list_sum_odd_help(List, Sum, 0, 0).

```

### **goal**

```

write("Len of [ ]: "),
list_length([], Length);
write("Len of [ 1, 2, 3 ]: "),
list_length([1, 2, 3], Length);

write("\nSum of [ ]: "),
list_sum([], Sum);
write("Sum of [ 1, 2, 3 ]: "),
list_sum([1, 2, 3], Sum);

write("\nSum of numbers in odd positions of [ ]: "),
list_sum_odd([], Sum);
write("Sum of numbers in odd positions of [ 1, 2, 3, 4, 5 ]: "),
list_sum_odd([1, 2, 3, 4, 5], Sum).

```

## Задание 2

Для одного из вариантов **ВОПРОСА** и одного из **заданий** составить **таблицу**, отражающую конкретный порядок работы системы:

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты! Для каждого запуска алгоритма унификации, требуется указать № выбранного правила и дальнейшие действия – и почему.

### Решение

**Вопрос:** `list_length([1, 2, 3], Len)`.

Шаг	Состояние резольвенты, и вывод	Сравниваемые термы; результат, подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	<code>list_length([1, 2, 3], Len)</code>	<code>list_length([1, 2, 3], Len)</code> и <code>list_length_help([], Sum, Sum)</code>  Неудача (разные главные функторы)	Прямой ход
2	...	...	...
3	<code>list_length([1, 2, 3], Len)</code>	<code>list_length([1, 2, 3], Len)</code> и <code>list_length(List, Len)</code>  Успех: <code>List = [1, 2, 3].</code>	Прямой ход
4	<code>list_length_help([1, 2, 3], Len, 0)</code>	<code>list_length_help([1, 2, 3], Len, 0)</code> и <code>list_length_help([], Sum, Sum)</code>  Неудача (неуспешно унифицируется первая пара соответствующих компонентов)	Прямой ход

5	list_length_help( [1, 2, 3], Len, 0)	list_length_help( [1, 2, 3], Len, 0) и list_length_help( [_ T], Len, Sum)  Успех: _ = 1, T = [2, 3], Sum = 0	Прямой ход
6	UpdSum = 0 + 1  list_length_help( [2, 3], Len, UpdSum)	Успех: UpdSum = 1	Прямой ход
7	list_length_help( [2, 3], Len, 1)	list_length_help( [2, 3], Len, 1) и list_length_help( [], Sum, Sum)  Неудача (неуспешно унифицируется первая пара соответствующих компонентов)	Прямой ход
8	list_length_help( [2, 3], Len, 1)	list_length_help( [2, 3], Len, 1) и list_length_help( [_ T], Len, Sum)  Успех: _ = 2, T = [3], Sum = 1	Прямой ход
9	UpdSum = 1 + 1  list_length_help( [3], Len, UpdSum)	Успех: UpdSum = 2	Прямой ход
10	list_length_help( [3], Len, 2)	list_length_help( [3], Len, 2) и list_length_help( [], Sum, Sum)  Неудача (неуспешно унифицируется первая пара соответствующих компонентов)	Прямой ход

11	list_length_help([3], Len, 2)	list_length_help([3], Len, 2) и list_length_help([_]T], Len, Sum)  Успех: _ = 3, T = [], Sum = 2	Прямой ход
12	UpdSum = 2 + 1  list_length_help([], Len, UpdSum)	Успех: UpdSum = 3	Прямой ход
13	list_length_help([], Len, 3)	list_length_help([], Len, 3) и list_length_help([], Sum, Sum)  Успех: Sum = 3, Len = 3	Прямой ход
14	!	Отсечение	Решение найдено: Len = 3  Откат: Len теряет значение
15	list_length([1, 2, 3], Len)	list_length([1, 2, 3], Len) и list_sum_help([], Sum, Sum)  Неудача (разные главные функторы)	Прямой ход
16-21	...	...	...
22	list_length([1, 2, 3], Len)	list_length([1, 2, 3], Len) и list_sum_odd(List, Sum)  Неудача (разные главные функторы)	Откат (конец БЗ)
23	Резольвента пуста. Найденные решения: Len = 3		