**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 | list\_length([1, 2, 3], Len) | list\_length([1, 2, 3], Len)  и  list\_length\_help([], S, S)  Неудача (разные главные функторы) | Прямой ход |
| 2 | … | … | … |
| 3 | list\_length([1, 2, 3], Len) | list\_length([1, 2, 3], Len)  и  list\_length(List, Len)  Успех:  List = [1, 2, 3]. | Прямой ход |
| 4 | list\_length\_help(  [1, 2, 3], Len, 0) | list\_length\_help(  [1, 2, 3], Len, 0)  и  list\_length\_help(  [], Sum, Sum)  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 |  |  |