



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 №11
по курсу «Функциональное и Логическое программирование»
на тему: «Рекурсия на Prolog»

Студент	<u>ИУ7-63Б</u>	_____	<u>Лагутин Д. В.</u>
	(Группа)	(Подпись, дата)	(Фамилия И. О.)
Преподаватель		_____	<u>Толпинская Н. Б.</u>
		(Подпись, дата)	(Фамилия И. О.)

Москва, 2023 г.

Задание

Используя хвостовую рекурсию, разработать (комментируя назначение аргументов) эффективную программу, 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);
- 4) Сформировать список из элементов числового списка, больших заданного значения;
- 5) Удалить заданный элемент из списка (один или все вхождения).
- 6) Объединить два списка.

```
1 domains
2     list_int = integer*.
3
4 predicates
5     list_len(list_int, integer).
6     list_len_inner(list_int, integer, integer).
7
8     list_sum(list_int, integer).
9     list_sum_inner(list_int, integer, integer).
10
11     list_sum_odd(list_int, integer).
12     list_sum_odd_sodd(list_int, integer, integer).
13     list_sum_odd_seven(list_int, integer, integer).
14
15     reverse(list_int, list_int).
16     reverse_inner(list_int, list_int, list_int).
17
18     list_gt(list_int, integer, list_int).
19     list_gt_inner(list_int, integer, list_int, list_int).
20
21     remove(list_int, integer, list_int).
22     remove_inner(list_int, integer, list_int, list_int).
23
24     merge(list_int, list_int, list_int).
25     merge_inner(list_int, list_int, list_int).
26
27 clauses
28     list_len(List, Len) :- list_len_inner(List, 0, Len).
```

```

29 list_len_inner([], Acc, Acc) :- !.
30 list_len_inner([_ | T], Acc, Len) :- AccIn = Acc + 1,
31                                     list_len_inner(T, AccIn, Len).
32
33 list_sum(List, Sum) :- list_sum_inner(List, 0, Sum).
34 list_sum_inner([], Acc, Acc) :- !.
35 list_sum_inner([H | T], Acc, Sum) :- AccIn = Acc + H,
36                                     list_sum_inner(T, AccIn, Sum).
37
38 list_sum_odd(List, Sum) :- list_sum_odd_seven(List, 0, Sum).
39 list_sum_odd_sodd([], Acc, Acc) :- !.
40 list_sum_odd_sodd([H | T], Acc, Sum) :- AccIn = Acc + H,
41                                     list_sum_odd_seven(T, AccIn,
42                                                         Sum).
43 list_sum_odd_seven([], Acc, Acc) :- !.
44 list_sum_odd_seven([_ | T], Acc, Sum) :- list_sum_odd_sodd(T, Acc,
45                                                         Sum).
46
47 reverse(List, Res) :- reverse_inner(List, [], Res).
48 reverse_inner([], Acc, Acc) :- !.
49 reverse_inner([H | T], Acc, Res) :- reverse_inner(T, [H | Acc], Res).
50
51 list_gt(List, Val, Res) :- list_gt_inner(List, Val, [], Inner),
52                             reverse(Inner, Res), !.
53 list_gt_inner([], _, Acc, Acc) :- !.
54 list_gt_inner([H | T], Val, Acc, Res) :- Val >= H,
55                                     list_gt_inner(T, Val, Acc,
56                                                         Res).
57 list_gt_inner([H | T], Val, Acc, Res) :- list_gt_inner(T, Val,
58                                                         [H | Acc], Res).
59
60 remove(List, Val, Res) :- remove_inner(List, Val, [], Inner),
61                             reverse(Inner, Res), !.
62 remove_inner([], _, Acc, Acc) :- !.
63 remove_inner([H | T], Val, Acc, Res) :- H = Val,
64                                     remove_inner(T, Val, Acc,
65                                                         Res).
66 remove_inner([H | T], Val, Acc, Res) :- remove_inner(T, Val,
67                                                         [H | Acc], Res).
68
69 merge(List1, List2, Res) :- reverse(List1, List1Tmp),
70                             merge_inner(List1Tmp, List2, ResRev),
71                             reverse(ResRev, Res).
72 merge_inner(List, [], List) :- !.
73 merge_inner(List, [H | T], Res) :- merge_inner([H | List], T, Res).

```

Таблица 1 – Вопрос: list_gt([-7, 9, -2, 5], -1, Res)

№ шага	Состояние резольвенты	Унифицируемые термы	Дальнейшие действия
1	list_gt([-7, 9, -2, 5], -1, Res).	list_gt([-7, 9, -2, 5], -1, Res) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
2	list_gt([-7, 9, -2, 5], -1, Res).	list_gt([-7, 9, -2, 5], -1, Res) = list_gt(List, Val, Res) Удача {List=[-7, 9, -2, 5], Val = -1, Res = Res}	Редукция
3	list_gt_inner([-7, 9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([-7, 9, -2, 5], -1, [], Inner) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
4	list_gt_inner([-7, 9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([-7, 9, -2, 5], -1, [], Inner) = list_gt_inner([], _, Acc, Acc) Неудача	Прямой ход. Переход к следующему знанию.
5	list_gt_inner([-7, 9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([-7, 9, -2, 5], -1, [], Inner) = list_gt_inner([H T], Val, Acc, Res) Удача {H = -7, T = [9, -2, 5], Val = -1, Acc = [], Inner = Res}	Редукция
6	-1 >= -7, list_gt_inner([9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	-1 >= -7 Удача	Прямой ход. Переход к следующему терму резольвенты
7	list_gt_inner([9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([-9, -2, 5], -1, [], Inner) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
8	list_gt_inner([9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([-7, 9, -2, 5], -1, [], Inner) = list_gt_inner([], _, Acc, Acc) Неудача	Прямой ход. Переход к следующему знанию.
9	list_gt_inner([9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([9, -2, 5], -1, [], Inner) = list_gt_inner([H T], Val, Acc, Res) Удача {H = 9, T = [-2, 5], Val = -1, Acc = [], Inner = Res}	Редукция
10	-1 >= 9, list_gt_inner([-2, 5], -1, [], Inner), reverse(Inner, Res), !.	-1 >= 9 Неудача	Откат

11	list_gt_inner([9, -2, 5], -1, [], Inner), reverse(Inner, Res), !.	list_gt_inner([9, -2, 5], -1, [], Inner) = list_gt_inner([H T], Val, Acc, Res) Удача {H = 9, T = [-2, 5], Val = -1, Acc = [], Inner = Res}	Редукция
12	list_gt_inner([-2, 5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([-2, 5], -1, [9], Inner) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
13	list_gt_inner([-2, 5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([-2, 5], -1, [9], Inner) = list_gt_inner([], _, Acc, Acc) Неудача	Прямой ход. Переход к следующему знанию.
14	list_gt_inner([-2, 5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([-2, 5], -1, [9], Inner) = list_gt_inner([H T], Val, Acc, Res) Удача {H = -2, T = [5], Val = -1, Acc = [9], Inner = Res}	Редукция
15	-1 >= -2, list_gt_inner([5], -1, [9], Inner), reverse(Inner, Res), !.	-1 >= -2 Удача	Прямой ход. Переход к следующему терму резольвенты
16	list_gt_inner([5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([-9, -2, 5], -1, [], Inner) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
17	list_gt_inner([5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([5], -1, [9], Inner) = list_gt_inner([], _, Acc, Acc) Неудача	Прямой ход. Переход к следующему знанию.
18	list_gt_inner([5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([5], -1, [9], Inner) = list_gt_inner([H T], Val, Acc, Res) Удача {H = 5, T = [], Val = -1, Acc = [9], Inner = Res}	Редукция
19	-1 >= 5, list_gt_inner([], -1, [9], Inner), reverse(Inner, Res), !.	-1 >= 5 Неудача	Откат
20	list_gt_inner([5], -1, [9], Inner), reverse(Inner, Res), !.	list_gt_inner([5], -1, [9], Inner) = list_gt_inner([H T], Val, Acc, Res) Удача {H = 5, T = [], Val = -1, Acc = [9], Inner = Res}	Редукция
21	list_gt_inner([], -1, [5, 9], Inner), reverse(Inner, Res), !.	list_gt_inner([], -1, [5, 9], Inner) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.

...
22	list_gt_inner([], -1, [5, 9], Inner), reverse(Inner, Res), !.	list_gt_inner([], -1, [5, 9], Inner) = list_gt_inner([], _, Acc, Acc) Удача { Acc = [5, 9], Acc = Inner }	... Редукция
23	!, reverse([5, 9], Res), !.	! Удача { }	Прямой ход. Переход к следующему терму резольвенты
24	reverse([5, 9], Res), !.	reverse([5, 9], Res) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
25	reverse([5, 9], Res), !.	reverse([5, 9], Res) = reverse(List, Res) Удача { List = [5, 9], Res = Res }	Редукция
26	reverse_inner([5, 9], [], Res), !	reverse_inner([5, 9], [], Res) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
27	reverse_inner([5, 9], [], Res), !.	reverse_inner([5, 9], [], Res) = reverse_inner([], Acc, Acc) Неудача	Прямой ход. Переход к следующему знанию.
28	reverse_inner([5, 9], [], Res), !.	reverse_inner([5, 9], [], Res) = reverse_inner([H T], Acc, Res) Удача { H = 5, T = [9], Acc = [], Res = Res }	Редукция
29	reverse_inner([9], [5], Res), !.	reverse_inner([9], [5], Res) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
30	reverse_inner([9], [5], Res), !.	reverse_inner([9], [5], Res) = reverse_inner([], Acc, Acc) Неудача	Прямой ход. Переход к следующему знанию.
31	reverse_inner([9], [5], Res), !.	reverse_inner([9], [5], Res) = reverse_inner([H T], Acc, Res) Удача { H = 9, T = [], Acc = [5], Res = Res }	Редукция
32	reverse_inner([], [9, 5], Res), !.	reverse_inner([], [9, 5], Res) = list_len(List, Len) Неудача	Прямой ход. Переход к следующему знанию.
...
33	reverse_inner([], [9, 5], Res), !.	reverse_inner([], [9, 5], Res) = reverse_inner([], Acc, Acc) Удача { Acc = [9, 5], Res = Acc }	Редукция

34	!, !.	!, Удача {}	! Удача {}	Прямой ход. Переход к следующему терму резольвенты
35	!.	!, Удача {}	! Удача {}	Прямой ход. Переход к следующему терму резольвенты
36				Резольвента пуста. Найдено решение. Откат
37	!.			В результате отката встречен предикат отсечения. Запрет использования list_gt. Откат.
38	!, !.			В результате отката встречен предикат отсечения. Запрет использования reverse_inner. Откат.
39	reverse([5, 9], Res), !.	reverse([5, 9], Res) = reverse_inner([], Acc, Acc) Неудача		Прямой ход. Переход к следующему знанию.
...
40	reverse([5, 9], Res), !.	reverse([5, 9], Res) = merge_inner(List, [H T], Res) Неудача		Откат. Достигнут конец базы знаний.
41	!, reverse([5, 9], Res), !.			В результате отката встречен предикат отсечения. Запрет использования list_gt_inner. Откат.
42	list_gt([-7, 9, -2, 5], -1, Res).	list_gt([-7, 9, -2, 5], -1, Res) = list_gt_inner([], _, Acc, Acc) Неудача		Прямой ход. Переход к следующему знанию.
...
43	list_gt([-7, 9, -2, 5], -1, Res).	list_gt([-7, 9, -2, 5], -1, Res) = merge_inner(List, [H T], Res) Неудача		Завершение работы. Достигнут конец базы знаний.