

# Формирование и модификация списков на Prolog

## 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).

Убедиться в правильности результатов. Для одного из вариантов ВОПРОСА и 1-ого задания составить таблицу, отражающую конкретный порядок работы системы.

## Код программы

```
include "lab18.inc"

domains
    intlist = integer*

predicates
    bigger_than(intlist, integer, intlist)
    odd_list(intlist, intlist)
    full_del(intlist, integer, intlist)
    set(intlist, intlist)

clauses
    bigger_than([Head | Tail], N, [Head | ResTail]) :- Head > N, !,
    bigger_than(Tail, N, ResTail).
    bigger_than([_ | Tail], N, Result) :- bigger_than(Tail, N, Result).
    bigger_than([], _, []).

    odd_list([_, Head | Tail], [Head | ResTail]) :- !, odd_list(Tail,
    ResTail).
    odd_list([], []).

    full_del([Head | Tail], N, [Head | ResTail]) :- Head <> N, !,
    full_del(Tail, N, ResTail).
    full_del([_ | Tail], N, Result) :- full_del(Tail, N, Result), !.
    full_del([], _, []).

    set([Head | Tail], [Head | Result]) :- full_del(Tail, Head, Nt), !,
    set(Nt, Result).
    set([], []).

goal
```

```
%bigger_than([1, 2, 4, 5], 3, Result).
    %Result=[4,5]
    %1 Solution
%odd_list([1, 2, 3, 4, 5, 6, 7, 8], Result).
    %Result=[2,4,6,8]
    %1 Solution
full_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
    %Result=[2,3,2,3,2,3]
    %1 Solution
%set([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).
    %Result=[1,2,3]
    %1 Solution
```

## Словесное описание порядка поиска ответа на вопрос

№ 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?)
0	bigger_than([1, 2, 4, 5], Result)		
1	1 > 3 ! bigger_than([2, 4, 5], 3, ResTail)	Сравнение: bigger_than([1, 2, 4, 5], 3, Result) = bigger_than([Head   Tail], N, [Head   ResTail]). Унификация успешна. Подстановка: {Head=1, Tail=[2, 4, 5]}, N=3, Result=[Head   ResTail]	Прямой ход, редукция резольвенты
2	! bigger_than([2, 4, 5], 3, ResTail)	1 > 3. Ложь	Откат относительно шага 1, прямой ход
3	bigger_than([2, 4, 5], 3, Result)	Сравнение: bigger_than([1, 2, 4, 5], 3, Result) = bigger_than([_   Tail], N, Result). Унификация успешна. Подстановка: {Tail=[2, 4, 5], N=3, Result=Result}	Прямой ход, редукция резольвенты
4	2 > 3 ! bigger_than(Tail=[4, 5], 3, ResTail)	Сравнение: bigger_than([2, 4, 5], 3, Result) = bigger_than([Head   Tail], N, [Head   ResTail]). Унификация успешна. Подстановка: {Head=2, Tail=[4, 5], N=3, Result=[Head   ResTail]}	Прямой ход, редукция резольвенты
5	! bigger_than(Tail=[4, 5], 3, ResTail)	2 > 3. Ложь	Откат относительно шага 3

№ 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?)
6	bigger_than([4, 5], 3, Result)	Сравнение: bigger_than([2, 4, 5], 3, Result) = bigger_than([_   Tail], N, Result). Унификация успешна. Подстановка: {Tail=[4, 5], N=3, Result=Result}	Прямой ход, редукция резольвенты
7	4 > 3 ! bigger_than(Tail=[5], 3, ResTail)	Сравнение: bigger_than([4, 5], 3, Result) = bigger_than([Head   Tail], N, [Head   ResTail]). Унификация успешна. Подстановка: {Head=4, Tail=[5]}, N=3, Result=[4   ResTail]	Прямой ход, редукция резольвенты
8	! bigger_than(Tail=[5], 3, ResTail)	4 > 3. Правда	Прямой ход, редукция резольвенты
9	bigger_than(Tail=[5], 3, ResTail)	!. Отсечение 7, 8	Прямой ход, редукция резольвенты
10	5 > 3 ! bigger_than([], 3, ResTail)	Сравнение: bigger_than(Tail=[5], 3, ResTail) = bigger_than([Head   Tail], N, [Head   ResTail]). Унификация успешна. Подстановка: {Head=5, Tail=[]}, N=3, Result=[5   ResTail]	Прямой ход, редукция резольвенты
11	! bigger_than([], 3, ResTail)	5 > 3. Правда	Прямой ход, редукция резольвенты
12	bigger_than([], 3, ResTail)	!. Отсечение 10, 11	Прямой ход, редукция резольвенты
13		Сравнение: bigger_than([], 3, ResTail) = bigger_than([Head   Tail], N, [Head   ResTail]). Унификация неуспешна	Прямой ход, следующее предложение
14	...	...	...
15		Сравнение: bigger_than([], 3, ResTail) = bigger_than([], _, []). Унификация успешна. Подстановка: {ResTail=[]}	<b>Вывод: Result=[4,5]</b>
16	...	...	...