



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное 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»

## **Лабораторная работа № 11 по дисциплине «Функциональное и логическое программирование»**

Тема Рекурсия в Prolog

Студент Шахнович Дмитрий Сергеевич

Группа ИУ7-62Б

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

Москва, 2025

# 1 Лабораторная работа

## 1.1 Задание

Используя хвостовую рекурсию разработать эффективную программу, позволяющую:

- 1) найти длину списка (по верхнему уровню);
- 2) найти сумму элементов числового списка;
- 3) найти сумму элементов числового списка, стоящих на нечётных позициях (нумерация с 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.

## 1.2 Программа

### 1.2.1 Код

```
domains
    intlist = integer*.
    num = integer.

predicates
    length(intlist , num).
    rlength(intlist , num, num).

    sum(intlist , num).
    rsum(intlist , num, num).

    oddsum(intlist , num).
    roddsum(intlist , num, num).

    listbigger(intlist , num, intlist).

    delone(intlist , num, intlist).
    delall(intlist , num, intlist).

    join(intlist , intlist , intlist).

clauses
```

```

rlength([], Length, Length) :- !.
rlength([_ | Tail], Length, Res) :- NewLen = Length + 1, rlength(Tail,
    NewLen, Res).
length(List, Length) :- rlength(List, 0, Length).

rsum([], Sum, Sum) :- !.
rsum([Elem | Tail], Sum, Res) :- NewSum = Sum + Elem, rsum(Tail, NewSum,
    Res).
sum(List, Sum) :- rsum(List, 0, Sum).

roddsum([], Sum, Sum) :- !.
roddsum([_], Sum, Sum) :- !.
roddsum([_, Elem | Tail], Sum, Res) :- NewSum = Sum + Elem, roddsum(Tail
    , NewSum, Res).
oddsom(List, Sum) :- roddsum(List, 0, Sum).

listbigger([], _, []) :- !.
listbigger([Elem | Tail], Big, [Elem | ResTail]) :- Elem > Big,
    listbigger(Tail, Big, ResTail), !.
listbigger([_ | Tail], Big, ResTail) :- listbigger(Tail, Big, ResTail).

delone([], _, []) :- !.
delone([Elem | Tail], Elem, Tail) :- !.
delone([Elem | Tail], Num, [Elem | ResTail]) :- delone(Tail, Num,
    ResTail).

delall([], _, []) :- !.
delall([Elem | Tail], Elem, ResTail) :- delall(Tail, Elem, ResTail), !.
delall([Elem | Tail], Num, [Elem | ResTail]) :- delall(Tail, Num,
    ResTail).

join([], List, List) :- !.
join([Elem | Tail], List, [Elem | ResTail]) :- join(Tail, List, ResTail)
.

```