



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 № 16  
по курсу «Функциональное и логическое программирование»  
на тему: «Рекурсия на Prolog»

Студент ИУ7-61Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Мицевич М. Д.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Толпинская Н. Б.  
(И. О. Фамилия)

2022 г.

## Задание

Используя хвостовую рекурсию, разработать программу, позволяющую найти

1.  $n!$ ,
2.  $n$ -е число Фибоначчи.

Убедиться в правильности результатов.

Для одного из вариантов ВОПРОСА и каждого задания 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!

### Листинг 1 – Решение задания №1

```
domains
val, res, accumulator = integer.
predicates
factorial(val, res).
factorial_tmp(val, res, accumulator).

fib(val, res).
fib_tmp(val, val, val, res).

clauses
factorial_tmp(N, Res, Acc) :- N > 1, N1 = N - 1, Acc1 = Acc * N,
    factorial_tmp(N1, Res, Acc1), !.
factorial_tmp(_, Res, Acc) :- Res = Acc.
factorial(N, Res) :- factorial_tmp(N, Res, 1).

fib_tmp(N, X1, X2, Res) :- N > 2, X11 = X2, X21 = X1 + X2, N1 = N
    - 1, fib_tmp(N1, X11, X21, Res), !.
fib_tmp(_, _, X2, Res) :- Res = X2.
fib(N, Res) :- fib_tmp(N, 1, 1, Res).

goal
%factorial(5, Res).
fib(5, Res).
```



Рисунок 1 – Таблица к заданию.

faNo 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?)
1	factorial(3, Res)	factorial_tmp(N, Res, Acc) и factorial(3, Res) <b>Результат:</b> неудача, термы не унифицируемы	Прямой ход. Переход к следующему предложению.
2	-//-	-//-	Прямой ход. Переход к следующему предложению.
3	factorial(3, Res), !	factorial(N, Res) и factorial(3, Res) <b>Результат:</b> успех + подстановка $\{N = 3, Res =$ $Res\}$	Заголовок правила заменяется его телом с учетом подстановки: factorial_tmp(3, Res, 1)
4	factorial_tmp(3, Res, 1)	factorial_tmp(N, Res, Acc) и factorial_tmp(3, Res, 1) <b>Результат:</b> успех + подстановка $\{N = 3, Res =$ $Res, Acc = 1\}$	Заголовок правила заменяется его телом с учетом подстановки: $3 > 1, NewN = 3 - 1,$ $NewAcc = 1 * 3,$ factorial_tmp(NewN, Res, NewAcc), !
5	$3 > 1, NewN = 3 - 1,$ $NewAcc = 1 * 3,$ factorial_tmp(NewN, Res, NewAcc), !	$3 > 1$ <b>Результат:</b> успех	Переход к следующему терму
6	$NewN = 3 - 1,$ $NewAcc = 1 * 3,$ factorial_tmp(NewN, Res, NewAcc), !	$NewN = 3 - 1$ <b>Результат:</b> успех + $\{NewN$ $= 2\}$	Переход к следующему терму с учётом подстановки $NewN = 2$
7	$NewAcc = 1 * 3,$ factorial_tmp(2, Res, NewAcc), !	$NewAcc = 1 * 3$ <b>Результат:</b> успех	Переход к следующему терму с учётом подстановки $NewAcc = 3$
8	factorial_tmp(2, Res, 3), !	factorial_tmp(2, Res, 3) и factorial_tmp(N, Res, Acc) <b>Результат:</b> успех + подстановка $\{N = 2, Res =$ $Res, Acc = 3\}$	Заголовок правила заменяется его телом с учетом подстановки: $2 > 1, NewN = 2 - 1,$ $NewAcc = 3 * 2,$ factorial_tmp(NewN, Res, NewAcc), !
9	$2 > 1, NewN = 2 - 1,$ $NewAcc = 3 * 2,$ factorial_tmp(NewN, Res, NewAcc), !, !	$2 > 1$ <b>Результат:</b> успех	Переход к следующему терму
6	$NewN = 2 - 1,$ $NewAcc = 3 * 2,$ factorial_tmp(NewN, Res, NewAcc), !, !	$NewN = 2 - 1$ <b>Результат:</b> успех + $\{NewN$ $= 1\}$	Переход к следующему терму с учётом подстановки $NewN = 1$
7	$NewAcc = 3 * 2,$	$NewAcc = 3 * 2$	Переход к следующему

	factorial_tmp(1, Res, NewAcc), !, !	<b>Результат:</b> успех	терму с учётом подстановки NewAcc = 6
8	factorial_tmp(1, Res, 6), !, !	factorial_tmp(1, Res, 6) и factorial_tmp(N, Res, Acc) <b>Результат:</b> успех + подстановка {N = 1, Res = Res, Acc = 6}	Заголовок правила заменяется его телом с учетом подстановки: 1 > 1, NewN = 1 - 1, NewAcc = 6 * 1, factorial_tmp(NewN, Res, NewAcc), !
9	1 > 1, NewN = 1 - 1, NewAcc = 6 * 1, factorial_tmp(NewN, Res, NewAcc), !, !, !	1 > 1 <b>Результат:</b> неудача	<b>Откат к 8</b>
10	factorial_tmp(1, Res, 6), !, !	factorial_tmp(_, Res, Acc) и factorial_tmp(1, Res, 6) <b>Результат:</b> успех + подстановка {_ = 1, Res = Res, Acc = 6}	Заголовок правила заменяется его телом с учетом подстановки: Res = 6
11	Res = 6, !, !	Res = 6 <b>Результат:</b> успех	Переход к следующему терму
12	!, !	! <b>Результат:</b> успех	Переход к следующему терму
13	!	! <b>Результат:</b> успех	Переход к следующему терму
14			Найдено решение Res=6 Отсечение => система больше не будет искать решение в процедуре factorial_tmp Откат к 3