



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»

## *Лабораторная работа №18*

*По предмету: «Функциональное и логическое  
программирование»*

Студент: Лаврова А. А.,  
Группа: ИУ7-65Б  
Преподаватель: Толпинская Н. Б.  
Строганов Ю. В.

Москва, 2020 г

### Задание:

Используя хвостовую рекурсию, разработать программу, позволяющую найти

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

Убедиться в правильности результатов.

### Листинг программы:

```
domains
```

```
    num = integer.
```

```
predicates
```

```
    factorial(num, num).
```

```
    factorial_aux(num, num, num).
```

```
    fib(num, num).
```

```
    fib_aux(num, num, num, num).
```

```
clauses
```

```
    factorial(0, 1) :- !.
```

```
    factorial(1, 1) :- !.
```

```
    factorial(N, R) :-  
        factorial_aux(N, 1, R).
```

```
    factorial_aux(1, R, R) :- !.
```

```
    factorial_aux(N, Acc, R) :-  
        NewN = N - 1,  
        NewAcc = Acc * N,  
        factorial_aux(NewN, NewAcc, R).
```

```
    fib(N, F) :-
```

```
        fib_aux(N, 0, 1, F).
```

```
    fib_aux(0, A, _, A) :- !.
```

```
    fib_aux(N, A, B, F) :-
```

```

N1 = N - 1,
Sum = A + B,
fib_aux(N1, B, Sum, F).

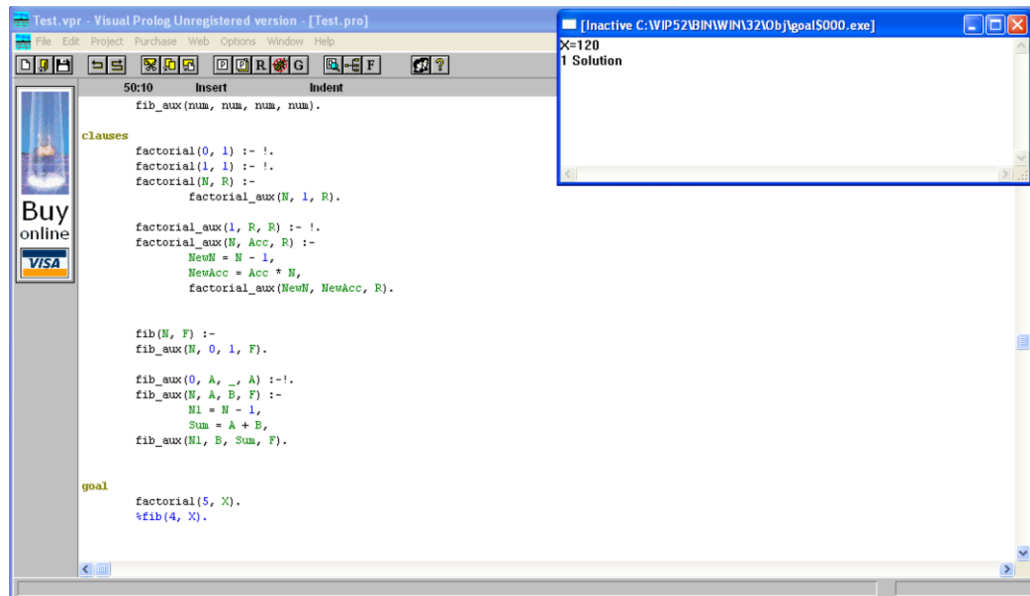
```

goal

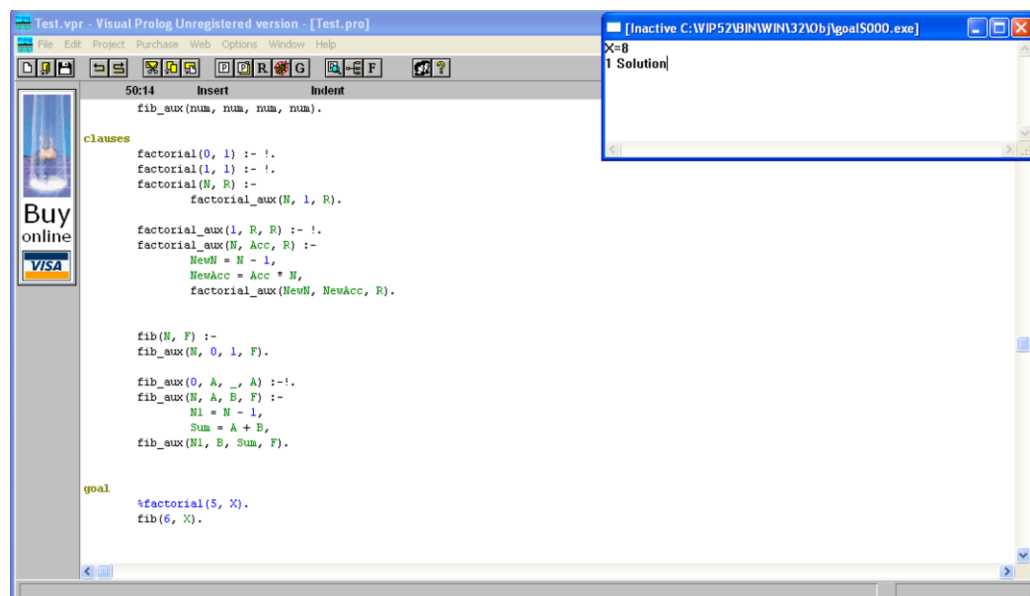
```
%factorial(5, X).
```

```
fib(4, X).
```

Результат работы программы:



Нахождение факториала числа.



Нахождение шестого числа Фибоначчи

Таблица для задания №1:

**factorial(5, X).**

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков <b>результат</b> (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	factorial(5, X)	$T1 = \text{factorial}(5, X)$ $T2 = \text{factorial}(0, 1)$  $5 \neq 0$ , унификация не произошла	Прямой ход
2	factorial(5, X)	$T1 = \text{factorial}(5, X)$ $T2 = \text{factorial}(1, 1)$  $5 \neq 1$ , унификация не произошла	Прямой ход
3	factorial(5, X)	$T1 = \text{factorial}(5, X)$ $T2 = \text{factorial}(N, R)$  Произошла унификация $N = 5, R = X$	Прямой ход
4	factorial_aux(5, 1, X).	$T1 = \text{factorial\_aux}(5, 1, X)$ $T2 = \text{factorial\_aux}(1, R, R)$  Унификация не произошла	Прямой ход
5	factorial_aux(5, 1, X).	$T1 = \text{factorial\_aux}(5, 1, X)$ $T2 = \text{factorial\_aux}(N, \text{Acc}, R)$  Произошла унификация, $N = 5, \text{Acc} = 1, R = X$	Прямой ход
6	$\text{NewN} = 5 - 1 = 4$ $\text{NewAcc} = \text{Acc} * N$	... (ряд неудачных попыток унификации)	Прямой ход

	$= 1 * 5 = 5$ factorial_aux(4, 5, R).		
7	NewN = $5 - 1 = 4$ NewAcc = Acc * N $= 1 * 5 = 5$ factorial_aux(4, 5, R). factorial_aux(4, 5, R).	T1 = factorial_aux(4, 5, R) T2 = factorial_aux(N, Acc, R)  Произошла унификация, N = 4, Acc = 5, R = R	Прямой ход
8	NewN = $N - 1 = 4 -$ $1 = 3$ NewAcc = Acc * N $= 5 * 4 = 20$ factorial_aux(3, 20, R). factorial_aux(4, 5, R). factorial_aux(4, 5, R).	<i>(до этого были  неудачные попытки  унификации, здесь и  далее пропущу для  экономии места)</i> T1 = factorial_aux(3, 20, R). T2= factorial_aux(N, Acc, R)  Произошла унификация, N = 3, Acc = 20, R = R	Прямой ход
9	NewN = $N - 1 = 3 -$ $1 = 2$ NewAcc = Acc * N $= 20 * 3 = 60$ factorial_aux(2, 60, R). factorial_aux(3, 20, R). factorial_aux(4, 5, R). factorial_aux(4, 5, R).	T1 = factorial_aux(2, 60, R). T2= factorial_aux(N, Acc, R)  Произошла унификация, N = 2, Acc = 60, R = R	Прямой ход
10	NewN = $N - 1 = 2 -$	T1 = factorial_aux(1, 120,	

	1 = 1 NewAcc = Acc * N = 60 * 2 = 120 factorial_aux(1, 120, R). factorial_aux(2, 60, R). factorial_aux(3, 20, R). factorial_aux(4, 5, R). factorial_aux(4, 5, R).	R). T2 = factorial_aux(1, R, R)  Унификация произошла успешно, R = 120	
	factorial_aux(2, 60, R). factorial_aux(3, 20, R). factorial_aux(4, 5, R). factorial_aux(4, 5, R).		Извлекаем из стека factorial_aux(1, 120, R).
	factorial_aux(3, 20, R). factorial_aux(4, 5, R). factorial_aux(4, 5, R).		Извлекаем из стека factorial_aux(2, 60, R).
	factorial_aux(4, 5, R). factorial_aux(4, 5, R).		Извлекаем из стека factorial_aux(3, 20, R).
	factorial_aux(4, 5, R).		Извлекаем из стека factorial_aux(4, 5, R).
			Извлекаем из стека

			factorial_aux(4, 5, R).
			Резольвента пуста. Вывод результата.

Таблица для задания №1:

**fib(3, X)**

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков <b>результат</b> (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	fib(3, X)	<i>(до этого были неудачные попытки унификации, здесь и далее пропущу для экономии места)</i> $T1 = \text{fib}(3, X)$ $T2 = \text{fib}(N, F)$  Произошла унификация, $N = 3, F = X$	Прямой ход
2	fib_aux(3, 0, 1, F)	$T1 = \text{fib\_aux}(3, 0, 1, F)$ $T2 = \text{fib\_aux}(N, A, B, F)$  Произошла унификация, $N = 3, A = 0, B = 1, F = F$	Прямой ход
3	$N1 = N - 1 = 3 - 1 = 2$ $\text{Sum} = A + B = 0 + 1 = 1$ fib_aux(2, 1, 1, F).	$T1 = \text{fib\_aux}(2, 1, 1, F).$ $T2 = \text{fib\_aux}(N, A, B, F)$  Произошла унификация, $N = 2, A = 1, B = 1, F = F$	Прямой ход
4	$N1 = N - 1 = 2 - 1 = 1$ $\text{Sum} = A + B = 1 + 1 = 2$	$T1 = \text{fib\_aux}(1, 1, 2, F).$ $T2 = \text{fib\_aux}(N, A, B, F)$  Произошла унификация,	Прямой ход

	fib_aux(1, 1, 2, F). fib_aux(2, 1, 1, F).	N = 1, A = 1, B = 2, F = F	
5	N1 = N - 1 = 1 - 1 = 0 Sum = A + B = 1 + 2 = 3 fib_aux(0, 2, 2, F). fib_aux(1, 1, 2, F). fib_aux(2, 1, 1, F).	T1 = fib_aux(0, 2, 2, F). T2 = fib_aux(0, A, _, A)  Произошла унификация, A = F = 2	Вывод результата. Обратный ход
6	fib_aux(1, 1, 2, F). fib_aux(2, 1, 1, F).		Извлекается из стека fib_aux(0, 2, 2, F).
7	fib_aux(2, 1, 1, F).		Извлекается из стека fib_aux(1, 1, 2, F).
			Извлекается из стека fib_aux(2, 1, 1, F).
			Резольвента пуста, завершение работы программы.

### Теоретическая часть

*1) Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?*

Рекурсия – это ссылка при описании объекта на уже описанный объект. При описании знания – ссылка на то же знание.

Для осуществления хвостовой рекурсии рекурсивный вызов определяемого предиката должен быть последней подцелью в теле рекурсивного правила и к моменту рекурсивного вызова не должно остаться точек возврата (непроверенных альтернатив). Параметры должны изменяться на каждом шаге так, чтобы в итоге либо сработал базис рекурсии, либо условие выхода из рекурсии, размещенное в самом правиле.

*2) Какое первое состояние резольвенты?*

Вопрос, который задается программе.

*3) В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)*



При наличии цели запускается процесс унификации. Также можно сказать, что унификация запускается тогда, когда резольвента не пуста.

*4) В каких пределах программы уникальны переменные?*

Переменные уникальны только в пределах предложения. Исключением является анонимные переменные.

*5) Как применяется подстановка, полученная с помощью алгоритма унификации?*

Применение подстановки  $\{X_1=T_1, \dots, X_n=T_n\}$  заключается в замене каждого вхождения переменной  $X_i$  на соответствующий терм  $T_i$ .

*6) Как меняется резольвента?*

Сначала из стека выбирается первая подцель, а затем замена подцели на тело подходящего правила. Потом к полученной конъюнкции применяется подстановка, то есть наибольший общий унификатор.

*7) В каких случаях запускается механизм отката?*

Откат происходит в случае тупиковой ситуации или в случае, если резольвента пуста.