



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 №18,19

По курсу: "Функциональное и Логическое программирование"

Тема _____ Рекурсия в прологе.
Группа _____ ИУ7-63Б
Студент _____ Сукочева А.
Преподаватель _____ Толпинская Н.Б.
Преподаватель _____ Строганов Ю. В.

Практическая часть л.р.16

Задание 1. Используя хвостовую рекурсию, разработать программу, позволяющую найти:

1. факториал числа;
2. n-ое число Фибоначчи.

DOMAINS

```
number = integer
```

PREDICATES

```
factorial(number, number, number).  
factorialWrapper(number, number).
```

CLAUSES

```
factorial(0, Res, Res) :- !.  
factorial(Number, Current, Res) :-  
    NextNumber = Number - 1,  
    Mult = Current * Number,  
    factorial(NextNumber, Mult, Res).  
  
factorialWrapper(Number, -1) :- Number < 0, !. % Error.  
factorialWrapper(Number, Res) :- factorial(Number, 1, Res).
```

GOAL

```
% factorialWrapper(5, Result).  
% factorialWrapper(-5, Result).  
factorialWrapper(0, Result).
```


```
DOMAINS
    number = integer

PREDICATES
    factorial(number, number, number).
    factorialWrapper(number, number).

CLAUSES
    factorial(0, Res, Res) :- !.
    factorial(Number, Current, Res) :-
        NextNumber = Number - 1,
        Mult = Current * Number,
        factorial(NextNumber, Mult, Res).

    factorialWrapper(Number, -1) :- Number < 0, !. % Error.
    factorialWrapper(Number, Res) :- factorial(Number, 1, Res).


GOAL
    factorialWrapper(5, Result).
```

 [Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=120
1 Solution


Рис. 1: Факториал числа

```
factorialWrapper(0, Result).
```

 [Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=1
1 Solution

```
factorialWrapper(-5, Result).
```

 [Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=-1
1 Solution

Рис. 3: Факториал числа

```
factorialWrapper(50, Result).
```

 [Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

PROGRAM ERROR. Module:OBJ\GOAL\$000.PRO Pos:224
Message:1031 Arithmetic overflow

Рис. 4: Факториал числа

```
DOMAINS
    number = integer

PREDICATES
    fibonacci(number, number, number, number).
    fibonacciWrapper(number, number).

CLAUSES
    fibonacci(1, Prev, _, Prev) :- !.
    fibonacci(Number, Prev, Current, Res) :-
        NewNumber = Number - 1,
        Newprev = Current,
        NewCurrent = Prev + Current,
        fibonacci(NewNumber, NewPrev, NewCurrent, Res).

    fibonacciWrapper(Number, -1) :- Number < 1, !. % Error.
    fibonacciWrapper(Number, Res) :- fibonacci(Number, 1, 1, Res).

GOAL
    % fibonacciWrapper(-15, Result).
    % fibonacciWrapper(1, Result).
    % fibonacciWrapper(2, Result).
    % fibonacciWrapper(3, Result).
    fibonacciWrapper(8, Result).
```

```
DOMAINS
    number = integer

PREDICATES
    fibonacci(number, number, number, number).
    fibonacciWrapper(number, number).

CLAUSES
    fibonacci(1, Prev, _, Prev) :- !.
    fibonacci(Number, Prev, Current, Res) :-
        NewNumber = Number - 1,
        Newprev = Current,
        NewCurrent = Prev + Current,
        fibonacci(NewNumber, NewPrev, NewCurrent, Res).

    fibonacciWrapper(Number, -1) :- Number < 1, !. % Error.
    fibonacciWrapper(Number, Res) :- fibonacci(Number, 1, 1, Res).

GOAL
    fibonacciWrapper(-15, Result).
```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=-1
1 Solution

Рис. 5: Фибоначчи

```
fibonacciWrapper(1, Result).
```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=1
1 Solution

```
fibonacciWrapper(8, Result).
```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=21
1 Solution

Рис. 6: Фибоначчи

Практическая часть л.р.17

Задание 2. 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 нечетных позициях исходного списка (нумерация от 0).

Убедиться в правильности результатов.

Длина списка.

```
DOMAINS
    list = integer*.

PREDICATES
    length(list, integer, integer).
    lengthWrapper(list, integer).

CLAUSES
    length([], Res, Res) :- !.
    length([_|T], CurrValue, Res) :-
        NewCurrValue = CurrValue + 1,
        length(T, NewCurrValue, Res).

    lengthWrapper(L, Res) :- length(L, 0, Res).

GOAL
    % lengthWrapper([1, 2, 3], Result).
    % lengthWrapper([], Result).
    lengthWrapper([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).
```

Сумма элементов числового списка.

```
DOMAINS
    list = integer*.

PREDICATES
    sum(list, integer, integer).
    sumWrapper(list, integer).

CLAUSES
    sum([], Res, Res) :- !.
    sum([H|T], CurrValue, Res) :-
        NewCurrValue = CurrValue + H,
        sum(T, NewCurrValue, Res).

    sumWrapper(L, Res) :- sum(L, 0, Res).

GOAL
    % sumWrapper([1, 2, 3], Result).
    % sumWrapper([], Result).
    sumWrapper([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).
```

Сумма элементов числового списка, стоящих на нечетных позициях.

```
DOMAINS
    list = integer*.
    flag = integer.

PREDICATES
    sum_odd(list, integer, integer, flag).
    sum_oddWrapper(list, integer).

CLAUSES
    sum_odd([], Res, Res, _) :- !.
    sum_odd([_|T], CurrValue, Res, 0) :- sum_odd(T, CurrValue, Res, 1).

    sum_odd([H|T], CurrValue, Res, 1) :-
        NewCurrValue = CurrValue + H,
        sum_odd(T, NewCurrValue, Res, 0).

    sum_oddWrapper(L, Res) :- sum_odd(L, 0, Res, 0).

GOAL
    % sum_oddWrapper([1, 2, 3], Result).
    %sum_oddWrapper([], Result).
    %sum_oddWrapper([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).
    sum_oddWrapper([1, 2, 3, 4, 5], Result).
    %sum_oddWrapper([1], Result).
```

10:36 Insert Indent

```
DOMAINS
    list = integer*.
    flag = integer.

PREDICATES
    sum_odd(list, integer, integer, flag).
    sum_oddWrapper(list, integer).

CLAUSES
    sum_odd([], Res, Res, _) :- !.
    sum_odd([_|T], CurrValue, Res, 0) :- sum_odd(T, CurrValue, Res, 1).

    sum_odd([H|T], CurrValue, Res, 1) :-
        NewCurrValue = CurrValue + H,
        sum_odd(T, NewCurrValue, Res, 0).

    sum_oddWrapper(L, Res) :- sum_odd(L, 0, Res, 0).

GOAL
    sum_oddWrapper([1, 2, 3], Result).
```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Result=2
1 Solution

Рис. 8: Сумма элементов числового списка, стоящих на нечетных позициях.

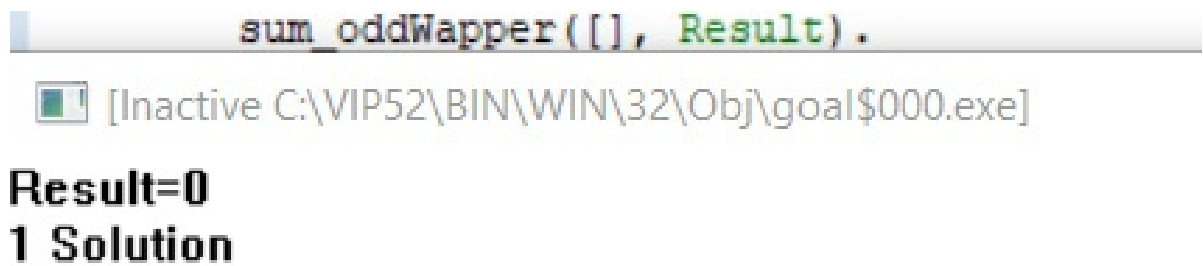


Рис. 9: Сумма элементов числового списка, стоящих на нечетных позициях.

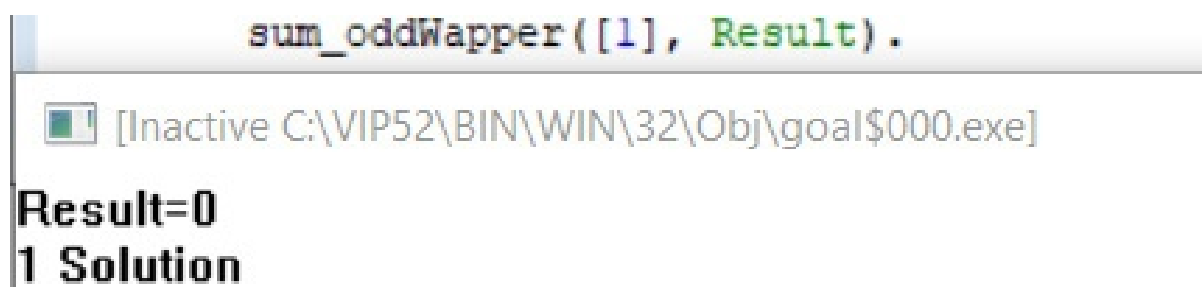


Рис. 10: Сумма элементов числового списка, стоящих на нечетных позициях.