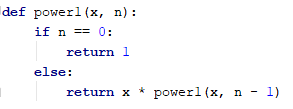
**Рекурсия**

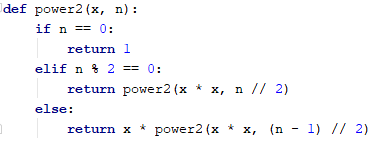
*Пример рекурсивной функции*

Рассмотрим 2 варианта функции возведения в степень.

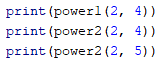
Первый вариант функции учитывает, что x0=1, xn=x\*xn-1:



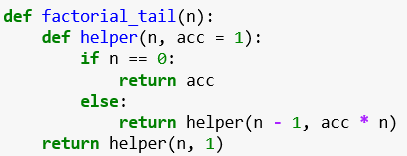
Во втором варианте используются другие соотношения: xm=(x2) m/2при четных m, xm=x\*(x2) (m-1)/2при нечетных m:



Использование функций:



*Пример хвостовой рекурсии*



*Задания для самостоятельной работы*

1. Реализовать функцию factorial(n) двумя способами: с помощью цикла и рекурсии.

1.1. С использованием глобальной переменной реализовать вывод на экран отладочной информации о вызовах функции factorial(n) печатающий с отступами, соответствующими глубине рекурсии.

Пример:

In: factorial(4)

factorial(4)

factorial(3)

factorial(2)

factorial(1)

factorial(0)

1.2. Реализовать функции printIn(s) и printOut(s), которые выводят строки s с отступами, при этом каждый вывод printIn(s) приводит к увеличению отступа, а каждый вывод printOut(s) приводит к уменьшению отступа.

1.3. С использованием printIn(s) и printOut(s) реализовать отладочный вывод работы factorial(n) как для вызовов функций, так и для возвращаемых значений.

Пример:

In: factorial(4)

factorial(4)

factorial(3)

factorial(2)

factorial(1)

factorial(0)

1

1

2

6

24

2. Рекурсивно реализовать функцию fib(n) вычисляющую значение n-го числа Фибоначи. С использованием printIn(s) и printOut(s) реализовать отладочный вывод работы fib(n)

3. Напишите функцию, которая переводит десятичное число n в p-ичную систему счисления. Цифры большие 9, как обычно, кодируются заглавными буквами латинского алфавита. Результатом функции является строка, соответствующая полученному числу.

4. Напишите рекурсивную функцию нахождения максимального значения среди элементов списка с индексами от L до R (включительно). В данном случае нужно использовать подход, который в литературе иногда называют «разделяй и властвуй». Его смысл сводится к тому, что при решении задачи большого размера мы находим решение нескольких задач меньшего размера, из которых получаем нужный результат. Если L = R, то решение очевидно.

5. Рассмотрим задачу получения всех возможных перестановок чисел от 1 до n (элементов А1, …,.Аn).

Для n=1 решение очевидно. Предположим, что мы можем переставить n-1 число с помощью функции Переставить(n-1). Тогда алгоритм решения этой задачи для n чисел (функцию Переставить(n)) можно описать, например, так:

* Переставить(n-1), оставляя Аn на месте.
* Для i от 1 до n-1 повторять {

Поменять местами Аn и Аi.

Переставить(n-1).

Вернуть Аn и Аi  на свои места.

}

Выведите на экран все перестановки зананного списка L.

6. Реализовать функцию подсчета n-го числа Фибоначчи с помощью хвостовой рекурсии.