

Лабораторная работа по теме «Рекурсивные функции»¹

Функция называется **рекурсивной**, если она сама себя вызывает. Задачи, которые решаются с использованием рекурсии, можно за один или несколько шагов свести к такой же задаче, но меньшей размерности.

Так же, как и у цикла, у рекурсии должно быть условие остановки (**базовый случай**). Всё решение сводится к решению базового случая. В случае, когда рекурсивная функция применяется для решения некоторой сложной задачи, не являющейся базовым случаем, выполняется некоторое количество шагов рекурсии (**рекурсивная ветвь**) до тех пор, пока не получим базовое решение.

Пример 1.

Рекурсивное вычисление факториала $n!$

```
def fact(n):
    if n == 1 or n == 0:          #базовый случай (конечная ветвь)
        return 1
    else:
        return n * fact(n-1) #шаг рекурсии (рекурсивная ветвь)
```

```
n = int(input("Введите n"))
print("%d! = %d" %(n, fact(n)))
```

Конечных и рекурсивных ветвей может быть несколько. Рекурсивные ветви сводят общую задачу к более простой задаче того же типа. Конечные ветви прекращают рекурсивные вызовы, когда задача сведена к базовому случаю.

Пример 2.

Вычисление элемента последовательности с заданным номером n по формуле:

$$a_n = 3 a_{n-1} + 5, \quad a_1 = 3.$$

```
def a(n):
    if n == 1:                    #базовый случай (конечная ветвь)
        return 3
    else:
        return 3 * a(n-1) + 5 #шаг рекурсии (рекурсивная ветвь)
```

```
n = int(input("Введите n"))
print("a[%d] = %d" %(n, a(n)))
```

Пример 3.

Проверка, состоит ли строка только из букв

```
def is_string_alpha(s):
    if len(s)<1:                  #базовый случай (конечная ветвь)
        return True
    else:
        return s[0].isalpha() and is_string_alpha(s[1:]) #шаг рекурсии (рекурсивная ветвь)
```

```
s = input("Введите строку: ")
print("Строка состоит из букв? - %s" %is_string_alpha(s))
```

¹Разработано А.М. Филимоновой (кафедра ВМиМФ мехмата ЮФУ)

Задачи для самостоятельного решения.

Решение всех задач необходимо оформить в виде рекурсивной функции.

1. Дано $x \in R, m \in N$. Вычислить x^m .
2. Числа в последовательности Фибоначчи вычисляются по формуле:

$$F_1 = 1, \quad F_2 = 1, \dots \quad F_k = F_{k-2} + F_{k-1}, \quad k = 3, 4, 5, \dots$$

Найти и вывести на экран элемент последовательности с номером k .

3. Дано натуральное число N . Проверить, является ли число N точной степенью двойки.
 4. Дано натуральное число N . Вычислить сумму его цифр.
 5. Вывести все числа от 1 до N в строку.
-
6. Дано два числа A и B . Вывести в одну строку все числа от A до B включительно в порядке возрастания, если $A < B$, или в порядке убывания в противном случае.
-
7. Для двух натуральных чисел m, n найти НОД(m, n), используя рекурсивный алгоритм Евклида.
-
8. Дана строка. Проверить, верно ли, что в строке есть хотя бы один из знаков арифметических операций: $+$, $-$, $*$, $/$
 9. Дана строка. Проверить, что она состоит только из заглавных латинских букв.
 10. Дана строка. Заменить все пробелы на символ `'*'`.
 11. Дана строка, состоящая только из букв. Проверить, является ли строка палиндромом.
 12. Дана строка S . Найти количество цифр в строке.
-
13. Дан список целых чисел a . Проверить, что все ли его элементы – чётные.
 14. Дан список целых чисел a . Проверить, верно ли, что в списке есть хотя бы одно отрицательное число.
 15. Дан список целых чисел a . Вычислить сумму всех элементов списка.
 16. Дан список целых чисел a , упорядоченный по возрастанию. Реализовать для такого списка алгоритм двоичного поиска, используя рекурсию.
-
17. Даны два целых неотрицательных числа m и n . Вычислить количество комбинаций из n разных элементов по m по формуле:

$$C_n^m = \begin{cases} 1, & m = 0 \text{ и } n > 0 \text{ или } m = n \geq 0 \\ 0, & m > n \geq 0 \\ C_{n-1}^{m-1} + C_{n-1}^m, & \text{иначе} \end{cases}$$

18. Даны три целочисленных списка разной длины, заполненных случайным образом. Найти максимальный элемент в каждом из списков, используя рекурсивную функцию.

19. Дан список целых чисел a длины N , заполненный случайным образом. Проверить, что числа в списке образуют невозрастающую последовательность.
-
20. Дано натуральное число N . Вычислить максимальную цифру в его восьмеричной записи.
21. Даны натуральные числа N и b ($2 \leq b \leq 16$). Реализовать перевод числа N из десятичной системы в систему счисления с основанием b .
22. Дано натуральное число N . Определить какой степени числа 5 является число N . В случае, если N не степень пятерки — выдать -1.
23. Даны два натуральных числа N и S . Проверить, совпадает ли сумма цифр числа N с числом S .
24. Дано натуральное число N . Посчитать количество делителей, отличных от 1 и самого числа N .