

Синтаксис Python. Функции, рекурсия.

■ Все задачи следует решить при помощи рекурсивных функций (разве что кроме задачи F).

Глубина стека вызовов — максимальное количество вызванных, но незаконченных (отложенных) функций в ходе выполнения программы. По умолчанию она равна 1000. Для изменения предельной глубины рекурсии используйте функцию `setrecursionlimit` из модуля `sys`.

Пример использования:

```
from sys import setrecursionlimit
...описания функций
setrecursionlimit(200000)
...текст программы
```

A. Сумма чисел

Дана последовательность чисел, завершающаяся числом 0. Найдите сумму всех этих чисел, не используя цикл и массивы.

Input	Output
1	6
2	
3	
0	

B. Разворот последовательности

Дана последовательность целых чисел, заканчивающаяся числом 0. Выведите эту последовательность в обратном порядке.

Решение этой задачи при помощи рекурсии позволяет обойтись без списков, строк и прочих структур данных для сохранения всех введённых чисел.

Input	Output
1	0
2	3
3	2
0	1

C° Алгоритм Евклида

Алгоритм Евклида для вычисления наибольшего общего делителя (gcd — greatest common divisor) двух натуральных чисел a и b ($a > b$) основан на следующих трёх фактах (для любых натуральных a и b):

$$gcd(a, b) = gcd(b, a)$$

$$gcd(a, 0) = a$$

$$gcd(a, b) = gcd(a - b, b)$$

Можно заметить, что многократно выполняемую операцию вычитания можно заменить операцией взятия остатка от деления:

$$gcd(a, b) = gcd(b, a \% b)$$

Рекомендуется выполнить соответствующие вычисления для таких, например, пар чисел: $a = 39, b = 11$ и $a = 14, b = 52$.

Обратите внимание на то, как изменяются числа в указанном выше соотношении, если первое число меньше второго. Например, чему равен остаток $14 \% 52$?

Если вы уловили идею, то функции `max` и `min` вам не понадобятся.

Даны два неотрицательных целых числа A и B . Требуется найти их наибольший общий делитель при помощи уже известного вам алгоритма Евклида.

Решение будет принято, только если рекурсивная функция, вычисляющая НОД, будет иметь ОДИН условный оператор.

На вход программе подаётся два целых неотрицательных числа. Программа должна вывести одно число — ответ на вопрос задачи.

Input	Output
12	4
16	

D. *Наименьшее общее кратное*

Наименьшее общее кратное (НОК) двух натуральных чисел — это наименьшее число, которое делится нацело на оба исходных числа. Напишите программу, которая вычисляет НОК двух данных натуральных чисел.

Input	Output
12 16	48

E° *Быстрое возведение в степень*

Дано вещественное число $a \neq 0$ и неотрицательное целое n ($n \leq 10^9$). Вычислите a^n .

Указание: воспользуйтесь тождествами $a^{2n} = (a^2)^n$ и $a^{2n+1} = a^{2n} \cdot a$

Input	Output
1.000000001 1000000000	2.7182820387553908

F* *Количество вызовов функции (числа Фибоначчи)*

Как известно, очередное число Фибоначчи равно сумме предыдущих двух. Первое и второе число Фибоначчи равны единице.

Программист Вася написал вычисление n -ого числа Фибоначчи с помощью рекурсивной функции, которая выглядит следующим образом:

```
def fibonacci(n):  
    if n < 3:  
        return 1  
    else:  
        return fibonacci(n - 1) + fibonacci(n - 2)
```

Сколько раз запустится эта функция прежде, чем будет получено значение?

На вход программе подаётся одно натуральное число. Программа должна вывести одно натуральное число — ответ на вопрос задачи. Гарантируется, что ответ не превосходит 10^{18} .

Input	Output
3	3
10	109
57	730870592323

G. *Сложение без сложения*

Напишите рекурсивную функцию $\text{sum}(a, b)$, возвращающую сумму двух целых неотрицательных чисел. Из всех арифметических операций допускаются только $+1$ и -1 . Циклы использовать нельзя.

На вход программе подаются два целых неотрицательных числа a и b . Программа должна вывести их сумму $a + b$.

Input	Output
2 2	4

H. *Фишки*

Дана полоска из клеток, пронумерованных от 1 до N . На каждом ходе разрешено поставить фишку на клетку (если её там еще нет) или снять фишку с клетки (если она там есть). При этом можно выбрать не любую клетку, а только клетку под номером 1 или клетку с номером на 1 больше, чем у первой (слева) занятой клетки.

Изначально полоска пуста. Требуется занять все клетки.

Программа должна вывести последовательность номеров клеток, с которыми совершается действие. Если фишка снимается, то номер клетки должен выводиться со знаком минус.

Количество действий не должно превышать 10^4 . Если существует несколько возможных решений задачи, то разрешается вывести любое.

Input	Output
3	1 2 -1 3 1

I. Ханойские башни

Головоломка «Ханойские башни» состоит из трех стержней, пронумерованных числами 1, 2, 3. На стержень 1 надета пирамидка из n дисков различного диаметра в порядке убывания диаметра (наверху самый маленький диск). Диски можно перекладывать с одного стержня на другой по одному, при этом диск нельзя класть на диск меньшего диаметра. Необходимо переложить всю пирамидку со стержня 1 на стержень 3 за минимальное число перекладываний.

Напишите программу, которая решает головоломку; для данного числа дисков n печатает последовательность перекладываний в формате $a\ b\ c$, где a — номер перекладываемого диска, b — номер стержня с которого снимается данный диск, c — номер стержня на который надевается данный диск.

Например, строка $1\ 2\ 3$ означает перемещение диска номер 1 со стержня 2 на стержень 3. В одной строке печатается одна команда. Диски пронумерованы числами от 1 до n в порядке возрастания диаметров.

Input	Output
2	1 1 2 2 1 3 1 2 3

J. Ремонт в Ханое

Решите задачу I (Ханойские башни) со следующим ограничением: запрещено перекладывать диски со стержня 1 на стержень 3 и наоборот.

Вам не нужно находить минимальное решение, но количество совершённых перемещений не должно быть больше 200000, при условии, что количество дисков не превосходит 10.

Input	Output
2	1 1 2 1 2 3 2 1 2 1 3 2 1 2 1 2 2 3 1 1 2 1 2 3

K. Циклические башни

Решите задачу I (Ханойские башни) со следующим ограничением: диск со стержня 1 можно перекладывать только на стержень 2, со стержня 2 на 3, а со стержня 3 на 1.

Вам не нужно находить минимальное решение, но количество совершённых перемещений не должно быть больше 200000, при условии, что количество дисков не превосходит 10.

Input	Output
2	1 1 2 1 2 3 2 1 2 1 3 1 2 2 3 1 1 2 1 2 3

L. Несправедливые башни

Решите задачу I (Ханойские башни) со следующим ограничением: запрещено класть самый маленький диск (номер 1) на **средний** колышек (номер 2).

Input	Output
2	1 1 3 2 1 2 1 3 1 2 2 3 1 1 3

M. Сортирующие башни

Решите задачу I (Ханойские башни) в такой формулировке: первоначально все диски лежат на стержне номер 1. Переместите диски с нечётными номерами на стержень номер 2, а с чётными номерами — на стержень номер 3.

Input	Output
3	1 1 2 2 1 3 1 2 3 3 1 2 1 3 2

Н. Обменные башни

Как и в предыдущих задачах, дано три стержня, на первом из которых надето n дисков различного размера. Необходимо их переместить на стержень 3 по следующим правилам: Самый маленький диск (номер 1) можно в любой момент переложить на любой стержень.

Перемещение диска номер 1 со стержня a на стержень b будем обозначать $1\ a\ b$.

Можно поменять два диска, лежащих на вершине двух стержней, если размеры этих дисков отличаются на 1. Например, если на вершине стержня с номером a лежит диск размером 5, а на вершине стержня с номером b лежит диск размером 4, то эти диски можно поменять местами.

Такой обмен двух дисков будем обозначать $0\ a\ b$ (указываются номера стержней, верхние диски которых обмениваются местами).

Для данного числа дисков n , не превосходящего 10, найдите решение головоломки. Вам не нужно находить минимальное решение, но количество совершённых перемещений не должно быть больше 200000.

Input	Output
2	1 1 3 0 1 3 1 1 3

О. Небоскрёб

В небоскрёбе n этажей. Известно, что если уронить стеклянный шарик с этажа номер p , и шарик разобьётся, то если уронить шарик с этажа номер $p + 1$, то он тоже разобьётся. Также известно, что при броске с последнего этажа шарик всегда разбивается.

Вы хотите определить минимальный номер этажа, при падении с которого шарик разбивается. Для проведения экспериментов у вас есть два шарика. Вы можете разбить их все, но в результате вы должны абсолютно точно определить этот номер.

Определите, какого числа бросков достаточно, чтобы заведомо решить эту задачу.

Программа получает на вход количество этажей в небоскрёбе n и выводит наименьшее число бросков, при котором можно всегда решить задачу.

Input	Output
4	2
20	6

Комментарий к первому примеру. Нужно бросить шарик со 2-го этажа. Если он разобьётся, то бросим второй шарик с 1-го этажа, а если не разобьётся — то бросим шарик с 3-го этажа.

Подсказки:

- Как следует действовать, если шарик был бы только один?
- Пусть шариков два и мы бросили один шарик с этажа номер k . Как мы будем действовать в зависимости от того, разобьётся ли шарик или нет?
- Пусть $f(n)$ — это минимальное число бросков, за которое можно определить искомый этаж, если бы в небоскрёбе было n этажей. Выразите $f(n)$ через значения $f(a)$ для меньших значений a .