**Якименко Олександр КМ-62**

[**Список квадратов**](http://pythontutor.ru/lessons/while/problems/list_of_squares/)

**По данному целому числу N распечатайте все квадраты натуральных   
чисел, не превосходящие N, в порядке возрастания.**  
n = int (input ())  
i=1  
while i\*\*2<=n:  
print(i\*\*2)   
i+=1

[**Минимальный делитель**](http://pythontutor.ru/lessons/while/problems/minimal_divisor/)

**Дано целое число, не меньшее 2. Выведите его наименьший натуральный  
делитель, отличный от 1.**  
  
n=int(input())  
k=2  
while n%k!=0:  
k+=1  
print(k)

[**Степень двойки**](http://pythontutor.ru/lessons/while/problems/powers_of_two/)

**По данному натуральному числу N найдите наибольшую целую степень   
двойки, не превосходящую N. Выведите показатель степени и саму   
степень.  
Операцией возведения в степень пользоваться нельзя!**  
  
n = int (input ())  
k=2  
m=1  
while k<=n:   
k\*=2  
m+=1  
print(m-1,k//2)

[**Утренняя пробежка**](http://pythontutor.ru/lessons/while/problems/running/)

**В первый день спортсмен пробежал x километров, а затем он каждый   
день увеличивал пробег на 10% от предыдущего значения. По данному   
числу y определите номер дня, на который пробег спортсмена  
составит не менее y километров.  
Программа получает на вход действительные числа x и y и должна  
вывести одно натуральное число.**  
  
x=int(input())  
y=int(input())  
k=1  
while x<y:  
x+=x\*0.1  
k+=1  
print(k)

[**Банковские проценты**](http://pythontutor.ru/lessons/while/problems/bank_interest/)

**Вклад в банке составляет x рублей. Ежегодно он увеличивается на p   
процентов, после чего дробная часть копеек отбрасывается.  
Определите, через сколько лет вклад составит не менее y рублей.  
Выражение «дробная часть копеек отбрасывается» означает, что если  
у вас оказалось 123.4567 рублей, т. е. 123 рубля и 45.67 копеек,  
то после округления у вас получится 123 рубля и 45 копеек, т.е.   
123.45 рублей.  
Программа получает на вход три натуральных числа: x, p, y и должна  
вывести одно целое число.**  
  
x = float(input(""))  
p = float(input(""))  
y = float(input(""))  
i = 0  
while x < y:  
x += x \* p\*0.01  
x = int(x\*100)/100  
i += 1  
print(i)

[**Длина последовательности**](http://pythontutor.ru/lessons/while/problems/seq_len/)

**Программа получает на вход последовательность целых неотрицательны-  
-х чисел, каждое число записано в отдельной строке.   
Последовательность завершается числом 0, при считывании которого   
программа должна закончить свою работу и вывести количество членов   
последовательности (не считая завершающего числа 0). Числа,   
следующие за числом 0, считывать не нужно.**  
i = 0  
while True:  
n = int(input(""))  
if n == 0:  
break  
i += 1  
print(i)

[**Сумма последовательности**](http://pythontutor.ru/lessons/while/problems/seq_sum/)

**Определите сумму всех элементов последовательности, завершающейся  
числом 0. В этой и во всех следующих задачах числа, следующие за  
первым нулем, учитывать не нужно.**  
  
sum=0  
while True:  
n = int(input(""))  
if n == 0:  
n+=n  
break  
sum+=n  
print(sum)

[**Среднее значение последовательности**](http://pythontutor.ru/lessons/while/problems/seq_avg/)

**Определите среднее значение всех элементов последовательности,   
завершающейся числом 0.**  
i=0  
sum=0  
while True:  
n = int(input(""))  
if n == 0:  
n+=n  
break  
sum+=n  
i+=1  
s=sum/i  
print(s)

[**Максимум последовательности**](http://pythontutor.ru/lessons/while/problems/seq_max/)

**Последовательность состоит из натуральных чисел и завершается   
числом 0. Определите значение наибольшего элемента последовательности.**  
i = 0  
max = 0  
while True:  
n = int(input(""))  
if n == 0:  
break  
if max<n:  
max =

[**Индекс максимума последовательности**](http://pythontutor.ru/lessons/while/problems/seq_index_of_max/)

**Последовательность состоит из натуральных чисел и завершается числом  
0. Определите индекс наибольшего элемента последовательности. Если   
наибольших элементов несколько, выведите индекс первого из них.**   
**Нумерация элементов начинается с нуля.**  
  
i = 0  
max = 0  
max\_i = 0  
while True:  
n = int(input(""))  
if n == 0:  
break  
if max<n:  
max = n  
max\_n = i  
i += 1  
print(max\_n)

[**Количество четных элементов последовательности**](http://pythontutor.ru/lessons/while/problems/seq_num_even/)

**Определите количество четных элементов в последовательности, завер-  
-шающейся числом 0.**  
  
i = 0  
kol=0  
while True:  
n = int(input(""))  
if n == 0:  
break  
if n%2==0:  
i += 1  
kol=i  
print(kol)

[**Количество элементов, которые больше предыдущего**](http://pythontutor.ru/lessons/while/problems/seq_increasing_neighbours/)

**Последовательность состоит из натуральных чисел и завершается   
числом 0. Определите, сколько элементов этой последовательности   
больше предыдущего элемента.**  
  
n = int(input())  
i = 0  
while n!= 0:  
n1 = int(input())  
if n1 != 0 and n < n1:  
i += 1  
n= n1  
print(i)

[**Второй максимум**](http://pythontutor.ru/lessons/while/problems/seq_second_max/)

**Последовательность состоит из различных натуральных чисел и   
завершается числом 0. Определите значение второго по величине   
элемента в этой последовательности. Гарантируется, что в   
последовательности есть хотя бы два элемента.**  
n=int(input())  
max=n  
max\_2=0  
while n!=0:  
n=int(input())  
if n>max:  
max\_2=max  
max=n  
elif n>max\_2 and n<max:  
max\_2=n  
print(max\_2)

[**Количество элементов, равных максимуму**](http://pythontutor.ru/lessons/while/problems/seq_num_maximal/)

**Последовательность состоит из натуральных чисел и завершается   
числом 0. Определите, сколько элементов этой последовательности   
равны ее наибольшему элементу.**max = 0   
max\_i = 0   
l = []   
k = 0   
while True:   
n = int(input())   
if n == 0:   
break   
l.append(n)   
if max<n:   
max = n   
for i in l:   
if i == max:   
k += 1   
print(k)

[**Числа Фибоначчи**](http://pythontutor.ru/lessons/while/problems/kth_fibonacci/)

**Последовательность Фибоначчи определяется так:   
φ0 = 0, φ1 = 1, φn = φn−1 + φn−2.   
По данному числу n определите n-е число Фибоначчи φn.   
Эту задачу можно решать и циклом for.**   
n = int(input(""))   
if n == 0:   
print(0)   
else:   
f = []   
f.append(0)   
f.append(1)   
i = 2   
while i<=n:   
f.append(f[i-1] + f[i-2])   
i += 1   
print(f[-1])

[**Номер числа Фибоначчи**](http://pythontutor.ru/lessons/while/problems/is_fibonacci/)

**Дано натуральное число A. Определите, каким по счету числом   
Фибоначчи оно является, то есть выведите такое число n, что   
φn = A. Если А не является числом Фибоначчи, выведите число -1.**   
x = int(input())   
f1 = 0   
f2 = 1   
n = 0   
fibo = 0   
while fibo != x:   
fibo = f1 + f2   
f2 = f1   
f1 = fibo   
n += 1   
if fibo > x:   
n = -1   
break   
print(n)

[**Максимальное число идущих подряд равных элементов**](http://pythontutor.ru/lessons/while/problems/seq_max_chunk_of_repetitions/)

**Дана последовательность натуральных чисел, завершающаяся числом 0.   
Определите, какое наибольшее число подряд идущих элементов этой   
последовательности равны друг другу.**   
i = 0   
max = 0   
n\_max = 0   
l = []   
l2 = [0]   
k = 1   
while True:   
n = int(input())   
if n == 0:   
break   
l.append(n)   
for i in range(1, len(l)):   
if l[i-1]== l[i]:   
k += 1   
else:   
l2.append(k)   
k = 1   
l2.append(k)   
l2.sort()   
print(l2[-1])

[**Стандартное отклонение**](http://pythontutor.ru/lessons/while/problems/std_dev/)

**Дана последовательность натуральных чисел x1x1, x2x2, ..., xnxn. Стандартным отклонением называется величина   
где s=x1+x2+…+xnns=x1+x2+…+xnn — среднее арифметическое последовательности.   
Определите стандартное отклонение для данной последовательности натуральных чисел, завершающейся числом 0.**

i = 1   
up = 0   
l = []   
suM = 0   
while True:   
x = int(input())   
if x == 0:   
break   
l.append(x)   
suM += x   
i += 1   
s = suM/(i-1)   
n = i-1   
for i in l:   
up += (i - s) \*\* 2   
print((up/(n-1)) \*\* 0.5)