Скорость первого туриста v1 км/ч, скорость второго туриста v2 км/ч. На какое расстояние друг от друга они отойдут через t часов, если будут двигаться в разные стороны. В первой строке скорость 1 туриста, во второй скорость 2 туриста, в третье строке t – количество часов.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 5  7  2 | 24 |
| 10  12  5 | 110 |

v1=int(input())

v2=int(input())

t=int(input())

print((v1\*t)+(v2\*t))

Длина Московской кольцевой автомобильной дороги — 109 километров. Байкер Вася стартует с нулевого километра МКАД и едет со скоростью vv километров в час. На какой отметке он остановится через tt часов?

Без использования условных операторов эту задачу существенно проще решить языке Python, чем на других языках программирования. Это связано с особенностями реализации целочисленных операций над отрицательными числами.

Входные данные

Программа получает на вход значение vv и tt. Если v>0v>0, то Вася движется в положительном направлении по МКАД, если же значение v<0v<0, то в отрицательном. (Гарантируется, что исходные числа - целые и находятся в промежутке от -1000 до +1000).

Выходные данные

Программа должна вывести целое число от 0 до 108 — номер отметки, на которой остановится Вася.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 60  2 | 11 |
| -1  1 | 108 |

vv=int(input())

tt=int(input())

print((vv\*tt)%109)

Дано двузначное натуральное число. Выведите на экран сумму его цифр.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 11 | 2 |
| 98 | 17 |

x = int(input())

print(x % 10 + x // 10)

Пирожок в столовой стоит aa рублей и bb копеек. Определите, сколько рублей и копеек нужно заплатить за nn пирожков.

Входные данные

Программа получает на вход три числа: aa, bb, nn - целые, положительные, не превышают 10000.

Выходные данные

Программа должна вывести два числа: стоимость покупки в рублях и копейках.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 10  15  2 | 20 30 |
| 2  50  4 | 10 0 |

aa=int(input())

bb=int(input())

nn=int(input())

print((aa\*nn)+((bb\*nn)//100),((bb\*nn)%100))

Дано целое положительное число X. Напишите программу, которая проверяет, является ли это число трехзначным.

Пояснение. Посмотрите деление на 100, 1000 и т. д.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 465 | YES |

x=int(input())

if x > 99 and x < 1000:

print('YES')

else:

print('NO')

n школьников делят k яблок поровну, неделяющийся остаток остается в корзинке. Сколько яблок останется в корзинке?

Входные данные

Программа получает на вход числа n и k - целые, положительные, не превышают 10000.

Выходные данные

Выведите ответ на задачу.

n=int(input())

k=int(input())

print(k%n)

Изучите работу функции type() выполнив следующие операторы. Значения типов должны быть выведены

print(type(None))

print(type(True))

print(type(False))

print(type(1))

print(type(5.3))

print(type(5 + 4j))

print(type([1, 5.3, False, 4]))

print(type((1, True, 3, 5+4j)))

print(type(range(5)))

print(type('Hello'))

print(type(b'a'))

print(type(bytearray([1,2,3])))

print(type(memoryview(bytearray('XYZ', 'utf-8'))))

print(type({'a', 3, True}) )

print(type(frozenset({1, 2, 3})) )

print(type({'a' : 32}))

| **Ожидаемый** | **Получено** |  |
| --- | --- | --- |
|  | <class 'NoneType'>  <class 'bool'>  <class 'bool'>  <class 'int'>  <class 'float'>  <class 'complex'>  <class 'list'>  <class 'tuple'>  <class 'range'>  <class 'str'>  <class 'bytes'>  <class 'bytearray'>  <class 'memoryview'>  <class 'set'>  <class 'frozenset'>  <class 'dict'> | <class 'NoneType'>  <class 'bool'>  <class 'bool'>  <class 'int'>  <class 'float'>  <class 'complex'>  <class 'list'>  <class 'tuple'>  <class 'range'>  <class 'str'>  <class 'bytes'>  <class 'bytearray'>  <class 'memoryview'>  <class 'set'>  <class 'frozenset'>  <class 'dict'> |

Дано два числа ***a*** и ***b***. Найдите гипотенузу треугольника с заданными катетами.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 3  4 | 3  4 | 3  4  5.0 |

import math

b=float(input())

a=float(input())

c=math.sqrt((a\*\*2+b\*\*2))

print(c)

Дано натуральное число. Выведите его последнюю цифру.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 179 | 179 | 179  9 |

n=int(input())

n1=n%10

print(n1)

Дано **трехзначное** число. Найдите сумму его цифр.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 179 | 179 | 179  17 |

n=int(input())

n3=n%10

n2=n//10%10

n1=n//100

print(n3+n2+n1)

*N* школьников делят *K* яблок поровну, неделящийся остаток остается в корзинке. Сколько яблок достанется каждому школьнику?

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 3  14 | 3  14 | 3  14  4 |

a=int(input())

b=int(input())

print(b//a)

В школе решили набрать три новых математических класса. Так как занятия по математике у них проходят в одно и то же время, было решено выделить кабинет для каждого класса и купить в них новые парты. За каждой партой может сидеть не больше двух учеников. Известно количество учащихся в каждом из трёх классов. Сколько всего нужно закупить парт чтобы их хватило на всех учеников? Программа получает на вход три натуральных числа: количество учащихся в каждом из трех классов.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 20  21  22 | 20  21  22 | 20  21  22  32 |

a=int(input())

b=int(input())

c=int(input())

print(a//2+b//2+c//2+a%2+b%2+c%2)

Написать программу, пишущую слово 'Python' 100 раз подряд. Сдайте на проверку программу, которая выводит эту строчку(только буквы, без кавычек).

print('Python'\*100)

Повторите следующие операторы, которые помогут вам разобраться с работой  ввода и вывода данных в Python:

print("Python is very simple language")

print('Hello')

print("""Python is very popular language.

It is also friendly language.""")

str1 = 'Wel'

print(str1,'come')

str1 = 'Welcome'

str2 = 'Python'

print(str1, str2)

str1 = 'Python'

str2 = ':'

print('Welcome' + str1 + str2)

| **Ожидаемый** | **Получено** |  |
| --- | --- | --- |
|  | Python is very simple language  Hello  Python is very popular language.  It is also friendly language.  Wel come  Welcome Python  WelcomePython: | Python is very simple language  Hello  Python is very popular language.  It is also friendly language.  Wel come  Welcome Python  WelcomePython: |

str1 = 'Python'

print("Welcome %s" % str1)

print("Actual Number = %d" %15)

print("Exponential equivalent of the number = %e" %15)

print("Float of the number = %f" %15)

print("Octal equivalent of the number = %o" %15)

print("Hexadecimal equivalent of the number = %x" %15)

str1 = 'World'

str2 = ':'

print("Python %s %s" %(str1,str2))

print("Welcome to %%Python %s" %'language')

print("Sunday\nMonday\nTuesday\nWednesday\nThursday\nFriday\nSaturday")

print("""

Language:

\t1 Python

\t2 Java\n\t3 JavaScript

""")

| **Ожидаемый** | **Получено** |  |
| --- | --- | --- |
|  | Welcome Python  Actual Number = 15  Exponential equivalent of the number = 1.500000e+01  Float of the number = 15.000000  Octal equivalent of the number = 17  Hexadecimal equivalent of the number = f  Python World :  Welcome to %Python language  Sunday  Monday  Tuesday  Wednesday  Thursday  Friday  Saturday  Language:  \t1 Python  \t2 Java  \t3 JavaScript | Welcome Python  Actual Number = 15  Exponential equivalent of the number = 1.500000e+01  Float of the number = 15.000000  Octal equivalent of the number = 17  Hexadecimal equivalent of the number = f  Python World :  Welcome to %Python language  Sunday  Monday  Tuesday  Wednesday  Thursday  Friday  Saturday  Language:  \t1 Python  \t2 Java  \t3 JavaScript |

*Напишите программу, которая возводит введенное число в степень 10.*

Входные данные

Входная строка содержит единственное вещественное число в диапазоне от 0 до 8.

Выходные данные

Нужно вывести 10-ю степень полученного числа с тремя знаками в дробной части.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 2.0 | 2.0 | 2.0  1024.000 |

a = float(input())

b = a\*a

c = b\*b

e = c\*c

g = e\*b

print ( "{:5.3f}".format(g) )

Выполните следующие операторы, которые помогут вам разобраться с работой  с типом float:

print("%f" % 5.1234567890)

print("%.5f" % 5.1234567890)

print("%9.5f" % 5.1234567890)

print("%015.5f" % 5.1234567890)

print("% 9f" % 5.1234567890)

print("% 9f" % -5.1234567890)

print("%-9.4f" % 5.1234567890)

Напишите программу, которая вводит трёхзначное число и разбивает его на цифры. Например, при вводе числа 123 программа должна вывести «1 2 3».

Входные данные

Входная строка содержит одно трёхзначное число.

Выходные данные

Программа должна вывести три цифры переданного ей числа, начиная с первой, разделённые пробелами.

Воспользоваться  методом join

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 123 | 123 | 123  1 2 3 |

a = int(input())

a = str(a)

print(int(a[0]),int(a[1]),int(a[2]))

foo = 1 # Задаем значения переменных

bar = 'bar' # Задаем значения переменных

baz = 3.14 # Задаем значения переменных

print('{}, {} and {}'.format(foo, bar, baz)) # Пары в скобках заменяются аргументами в порядке их передачи "1, bar and 3.14"

print('{0}, {1}, {2}, and {1}'.format(foo, bar, baz)) # Номера соответствуют индексам аргументов : "1, bar, 3.14, and bar"

print('{2}, {0}, {2}, and {1}'.format(foo, bar, baz)) # Номера соответствуют индексам аргументов

#print('{0}, {1}, {2}, and {3}'.format(foo, bar, baz)) выдает ошибку так как индекс 3 не существует

print("X value is: {xx}. Y value is: {yyl}.".format(xx=2, yyl=3)) # Можно использовать именованные аргументы

# аргументы по умолчанию print("Hello {}, your balance is {}.".format("Adam", 230.2346))

# позиционные аргументы print("Hello {0}, your balance is {1}.".format("Adam", 230.2346))

# аргументы ключевые слова print("Hello {name}, your balance is {blc}.".format(name="Adam", blc=230.2346))

# смешанные аргументы print("Hello {0}, your balance is {blc}.".format("Adam", blc=230.2346))

print('{:~^20}'.format('centered')) #вы можете также включать в спецификацию формата внутри фигурных скобок. #Это выражение , которое следует , и особые правила должны предшествовать двоеточие ( : ). # :~^20 ( ^ означает выравнивание по центру, общая ширина 20, подстановка символа ~ )

# целочисленные аргументы

print("The number is:{:d}".format(123))

# аргументы с плавающей точкой

print("The float number is:{:f}".format(123.4567898))

# восьмеричный, двоичный и шестнадцатеричный формат

print("bin: {0:b}, oct: {0:o}, hex: {0:x}".format(12))

# целые числа с выравниванием по правому краю

print("{:5d}".format(12))

# числа с плавающей точкой с выравниванием по центру

print("{:^10.3f}".format(12.2346))

# выравнивание целого числа по левому краю заполнено нулями

print("{:<05d}".format(12))

# числа с плавающей точкой с выравниванием по центру

print("{:=8.3f}".format(-12.2346))

# показать знак +

print("{:+f} {:+f}".format(12.23, -12.23))

# показать знак -

print("{:-f} {:-f}".format(12.23, -12.23))

# показать место для знака +

print("{: f} {: f}".format(12.23, -12.23))

# отступ строки с выравниванием по левому краю

print("{:5}".format("cat"))

# отступ строки с выравниванием по правому краю

print("{:>5}".format("cat"))

# заполнение строк с выравниванием по центру

print("{:^5}".format("cat"))

# заполнение строк с выравниванием по центру # и '\*' - символ заполнения

print("{:\*^5}".format("cat"))

| **Ожидаемый** | **Получено** |  |
| --- | --- | --- |
|  | 1, bar and 3.14  1, bar, 3.14, and bar  3.14, 1, 3.14, and bar  X value is: 2. Y value is: 3.  ~~~~~~centered~~~~~~  The number is:123  The float number is:123.456790  bin: 1100, oct: 14, hex: c  12  12.235  12000  - 12.235  +12.230000 -12.230000  12.230000 -12.230000  12.230000 -12.230000  cat  cat  cat  \*cat\* | 1, bar and 3.14  1, bar, 3.14, and bar  3.14, 1, 3.14, and bar  X value is: 2. Y value is: 3.  ~~~~~~centered~~~~~~  The number is:123  The float number is:123.456790  bin: 1100, oct: 14, hex: c  12  12.235  12000  - 12.235  +12.230000 -12.230000  12.230000 -12.230000  12.230000 -12.230000  cat  cat  cat  \*cat\* |

*Напишите программу, которая находит сумму, произведение и среднее арифметическое трёх целых чисел, введённых с клавиатуры.*

Входные данные

Три целых числа вводятся c новой строчки.

Выходные данные

Программа должна вывести сумму, произведение и среднее арифметическое введенных чисел. Среднее арифметическое нужно вывести с точностью 3 знака после десятичной точки.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 4  5  7 | 4  5  7 | 4  5  7  4+5+7=16  4\*5\*7=140  (4+5+7)/3=5.333 |

x=int(input())

y=int(input())

z=int(input())

print("{}+{}+{}={}".format(x, y, z, (x+y+z)))

print(f"{x}\*{y}\*{z}={x\*y\*z}")

print("(%d+%d+%d)/3=%.3f" % (x, y, z, (x+y+z)/3))

Напишите программу, которая вводит координаты двух точек на числовой оси и выводит расстояние между ними.

Входные данные Данные будут типа float !

В первой строке вводятся через пробел координаты первой точки сначала *x*-координата,  *y*-координата, во второй строке в том же порядке – координаты второй точки. Все координаты – вещественные числа.

Выходные данные

Программа должна вывести одно число: расстояние между точками с тремя знаками в дробной части.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 1 1  2 2 | 1 1  2 2  1.414 |

x1, y1 = map(float, input().split())

x2, y2 = map(float, input().split())

f = ((x1 - x2) \*\* 2 + (y1 - y2) \*\* 2) \*\* 0.5

print(f'{f:.3f}')

Найдите максимум из двух чисел.

Входные данные

Даны два целых числа, каждое число записано в отдельной строке.

Выходные данные

Выведите наибольшее из данных чисел.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 1  2 | 1  2 | 1  2  2 |

a = int(input())

b = int(input())

if a > b:

print(a)

else:

print(b)

Требуется определить, является ли данный год високосным. (Напомним, что год является високосным, если его номер кратен 4, но не кратен 100, а также если он кратен 400.)

Входные данные

Вводится единственное число - номер года (целое, положительное, не превышает 30000).

Выходные данные

Требуется вывести слово YES, если год является високосным и NO - в противном случае.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 2007 | 2007 | 2007  NO |
| 2000 | 2000 | 2000  YES |

a = int(input())

if a%4 == 0 and (a%100!=0 or a%400==0):

print("YES")

else:

print("NO")

В математике функция sign(x) (знак числа) определена так:  
sign(x) = 1, если x > 0,  
sign(x) = -1, если x < 0,  
sign(x) = 0, если x = 0.

Для данного числа x выведите значение sign(x). Эту задачу желательно решить с использованием каскадных инструкций if... elif... else

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 1534 | 1534 | 1534  1 |
| -42 | -42 | -42  -1 |
| 0 | 0 | 0  0 |

a = int(input())

if a > 0:

print(1)

elif a < 0:

print(-1)

elif a == 0:

print(0)

Даны три целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 3 (если все совпадают), 2 (если два совпадает) или 0 (если все числа различны).

Входные данные

Вводятся три целых числа.

Выходные данные

Выведите ответ на задачу.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 1  2  3 | 1  2  3 | 1  2  3  0 |

a = int(input())

b = int(input())

c = int(input())

if a == b == c:

print(3)

elif a==b or b==c or c==a:

print(2)

else:

print(0)

Напишите программу, которая вводит с клавиатуры номер месяца и определяет, сколько дней в этом месяце. При вводе неверного номера месяца должно быть выведено сообщение об ошибке. Считается, что год невисокосный.

*Входные данные*

 Входная строка содержит единственное целое число – номер месяца (возможно, неправильный).

*Выходные данные*

 Выходная строка должна содержать одно целое число – количество дней в этом месяце или 0, если был введён неверный номер месяца.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 4 | 4 | 4  30 |

a=int(input())

s = (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)

if a > 12 or a <= 0:

print(0)

if a > 0 and a < 13:

print(s[a - 1])

Напишите программу, которая вводит номер месяца и выводит название времени года на английском языке. При вводе неверного номера месяца должно быть выведено слово 'NO'.

Входные данные

Входная строка содержит единственное число – номер месяца (возможно, неверный).

Выходные данные

Нужно вывести название времени года (на английском языке), соответствующее введённому номеру месяца ('winter' – зима, 'spring' – весна, 'summer' – лето, 'autumn' – осень). Если введён неверный номер месяца, программы должна вывести слово 'NO'.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 4 | 4 | 4  spring |
| 25 | 25 | 25  NO |

a = int(input())

s = ('winter', 'spring', 'summer', 'autumn')

if a > 0 and a < 3 or a == 12:

print(s[0])

elif a > 2 and a < 6:

print(s[1])

elif a > 5 and a < 9:

print(s[2])

elif a > 8 and a < 12:

print(s[3])

else: print("NO")

Напишите программу на Python для вычисления площади правильного многоугольника. Обратите внимание на подключение математического пакета Math.

from math import tan, pi

n\_sides = int(input())

s\_length = float(input())

p\_area = n\_sides \* (s\_length \*\* 2) / (4 \* tan(pi / n\_sides))

print("The area of the polygon is: ",p\_area)

Изучите работу модуля decimal. Напишите  программу Python, чтобы получить квадратный корень и экспоненту данного десятичного числа.

**Для примера:**

| **Результат** |
| --- |
| Square root of 1.44 is : 1.2  exponential of 1.44 is : 4.220695816996552825673328929 |

from decimal import \*

x = Decimal('1.44')

print("Square root of ",x, " is :", x.sqrt())

print("exponential of ",x, " is :", x.exp())

Напишите программу Python для преобразования полярных координат в прямоугольные координаты .

import cmath

cn = complex(3,4)

# получение значений полярных координат

print("Polar Coordinates: ",cmath.polar(cn))

cn1 = cmath.rect(2, cmath.pi)

print("Polar to rectangular: ",cn1)

Реализовать программу  для преобразования числа с плавающей запятой в неправильную дробь.

from fractions import Fraction

value = 4.2

print(Fraction(value).limit\_denominator())

Реализовать программу Python для сложения, вычитания, умножения и деления двух дробей.

**Для примера:**

| **Результат** |
| --- |
| 2/3 + 3/7 = 23/21  2/3 - 3/7 = 5/21  2/3 \* 3/7 = 2/7  2/3 / 3/7 = 14/9 |

import fractions

f1 = fractions.Fraction(2, 3)

f2 = fractions.Fraction(3, 7)

print('{} + {} = {}'.format(f1, f2, f1 + f2))

print('{} - {} = {}'.format(f1, f2, f1 - f2))

print('{} \* {} = {}'.format(f1, f2, f1 \* f2))

print('{} / {} = {}'.format(f1, f2, f1 / f2))

Написать программу выводящую число пи. Указание: необходимо использовать математический пакета math.

from math import pi

print(pi)

Почитать случайное число в промежутке от 0 до 1

import random

def xx():

return print(random.random())

Получить случайное целое число в промежутке от 2 до 4

import random

def xx(a,b):

return random.randint(a,b)

Даны два целых числа A и B (при этом A ≤ B). Выведите все числа от A до B включительно.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 1  10 | 1  10  1 2 3 4 5 6 7 8 9 10 |

a = int(input())

b = int(input())

print(\*list(range(a, b + 1)))

или

a = int(input())

b = int(input())

for i in range(a, b + 1) :

print (i, end = " ")

Даны два целых числа A и В. Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае.

Входные данные

Вводятся два целых числа.

Выходные данные

Выведите ответ на задачу.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 1  10 | 1  10  1 2 3 4 5 6 7 8 9 10 |

a = int(input())

b = int(input())

if a < b:

print(\*list(range(a, b + 1)))

else:

print(\*list(range(a, b - 1, -1)))

Даны два целых числа A и В, A>B. Выведите все нечётные числа от A до B включительно, в порядке убывания. В этой задаче можно обойтись без инструкции if.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 7  1 | 7  1 | 7  1  7 5 3 1 |

a = int(input())

b = int(input())

for i in range(a - (a + 1) % 2, b - b % 2, -2):

print(i, end=' ')

Входные данные

Вводятся целые числа *a* и *b*. Гарантируется, что *a* не превосходит *b*

Выходные данные

Выведите (через пробел) все четные числа от *a* до *b* (включительно).

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 2  5 | 2  5 | 2  5  2 4 |

a = int(input())

b = int(input())

for i in range(a,b+1):

if i%2==0:

print(i,end=' ')

Напишите программу, которая вводит два целых числа и находит их произведение, не используя операцию умножения. Учтите, что числа могут быть отрицательными.

Входные данные

Входная строка содержит два целых числа.

Выходные данные

Программа должна вывести произведение введённых чисел.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 6  12 | 6  12 | 6  12  72 |

a = int(input())

b = int(input())

if a<0 and b<0:

a=-a

b=-b

elif b<0:

a,b=b,a

s=0

for i in range(b):

s=s+a

print(s)

Входные данные

Вводятся целые числа *a* и *b*. Гарантируется, что *a* не превосходит *b*.

Выходные данные

Выведите все числа на отрезке от *a* до *b*, являющиеся полными квадратами. Если таких чисел нет, то ничего выводить не нужно.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 2  8 | 2  8 | 2  8  4 |
| 17  164 | 17  164 | 17  164  25  36  49  64  81  100  121  144 |

from math import sqrt

a = int(input())

b = int(input())

for i in range(a, b+1):

if sqrt(i).is\_integer()==True:

print(i)

Факториалом числа n называется произведение 1 × 2 × ... × n. Обозначение: n!.

res = 1

n = int(input())

for i in range(1, n + 1):

res \*= i

print(res)

Входные данные

Вводится число *N*, а затем *N* чисел.

Выходные данные

Подсчитайте и выведите, сколько среди данных *N* чисел нулей.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4  9  0  8  1 | 4  9  0  8  1  1 |

X = int(input())

s = 0

for i in range(X):

a = int(input())

if a==0:

s = s + 1

print (s)

Выведите все натуральные делители числа *x* в порядке возрастания (включая 1 и само число).

Входные данные

Вводится натуральное число *x*

Выходные данные

Выведите все делители числа *x*

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 32 | 32  1 2 4 8 16 32 |

x = int(input())

for i in range(1,x+1):

if x % i == 0:

print(i,end = ' ')

Напишите программу, которая считает сумму цифр введённого числа.

Входные данные

Входная строка содержит одно натуральное число.

Выходные данные

Программа должна вывести сумму цифр введённого числа.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 12345 | 12345  15 |

s=int(input())

m=0

while s!=0:

m+=s%10

s=s//10

print(m)

или

a = input()

kol = 0

for i in a:

kol += int(i)

print(kol)

Определите сумму всех элементов последовательности, завершающейся числом 0.

Входные данные

Вводится последовательность целых чисел, оканчивающаяся числом 0 (само число 0 в последовательность не входит, а служит как признак ее окончания).

Выходные данные

Выведите ответ на задачу.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 1  7  9  0 | 1  7  9  0  17 |

a = int(input())

sm = 0

while a != 0:

sm += a

a = int(input())

print(sm)

Последовательность состоит из натуральных чисел, не превосходящих 109, и завершается числом 0. Определите значение наибольшего элемента последовательности.

Входные данные

Вводится последовательность целых чисел, оканчивающаяся числом 0 (само число 0 в последовательность не входит, а служит как признак ее окончания).

Выходные данные

Выведите ответ на задачу.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 1  7  9  0 | 1  7  9  0  9 |

max = 0

element = -1

while element != 0:

element = int(input())

if element > max:

max = element

print(max)

По данному числу nn определите nn-е число Фибоначчи .

Входные данные

Вводится натуральное число n.

Выходные данные

Выведите ответ на задачу

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 6 | 6  8 |

n = int(input())

x, y = 0, 1

if n < 2:

print(n)

else:

while n > 1:

t = y

y = y + x

x = t

n -= 1

print(y)

Последовательность состоит из натуральных чисел и завершается числом 0. Определите, сколько элементов этой последовательности больше предыдущего элемента.

Входные данные

Вводится последовательность целых чисел, оканчивающаяся числом 0 (само число 0 в последовательность не входит, а служит как признак ее окончания).

Выходные данные

Выведите ответ на задачу.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 1  7  9  0 | 1  7  9  0  2 |

a = int(input())

b = 0

while a != 0:

c = int(input())

if c != 0 and a < c:

b += 1

a = c

print(b)

Напишите программу, которая строит "квадрат" из знаков '\*' заданного размера. Используйте процедуру.

Входные данные

Входная строка содержит единственное натуральное число – длину стороны квадрата *N*.

Выходные данные

Программа должна вывести заполненный квадрат размером *N*× *N*, состоящий из знаков '\*'.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 5 | 5  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\* |

N = int(input())

a = '\*'

n = N

while n>0:

print (a\*N)

n -= 1

На вход программе подается натуральное число N. Напишите программу, которая находит сумму квадратов всех чисел от 1 до N

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 2 | 5 |
| 10 | 385 |

n = int(input())

s=0

for i in range(1, n+1) :

s=s+i\*\*2

print(s)

Выведите на экран таблицу умножения чисел от 1 до 9.

**Для примера:**

| **Результат** |
| --- |
| 1 2 3 4 5 6 7 8 9  2 4 6 8 10 12 14 16 18  3 6 9 12 15 18 21 24 27  4 8 12 16 20 24 28 32 36  5 10 15 20 25 30 35 40 45  6 12 18 24 30 36 42 48 54  7 14 21 28 35 42 49 56 63  8 16 24 32 40 48 56 64 72  9 18 27 36 45 54 63 72 81 |

for i in range(1, 9 + 1): # строка

for j in range(1, 9 + 1): # столбец

print(j\*i, end = ' ')

print()# переход на ноую строку

Выведите на экран таблицу сложения чисел от 1 до 9, т.е. каждый элемент таблицы это значение суммы индекса строки и столбца.

**Для примера:**

| **Результат** |
| --- |
| 2 3 4 5 6 7 8 9 10  3 4 5 6 7 8 9 10 11  4 5 6 7 8 9 10 11 12  5 6 7 8 9 10 11 12 13  6 7 8 9 10 11 12 13 14  7 8 9 10 11 12 13 14 15  8 9 10 11 12 13 14 15 16  9 10 11 12 13 14 15 16 17  10 11 12 13 14 15 16 17 18 |

for i in range(1, 9 + 1): # строка

for j in range(1, 9 + 1): # столбец

print(j+i, end = ' ')

print()# переход на ноую строку

Выведите на экран таблицу степеней числа a в степени b, где a и b - числа от 1 до 5, т.е. каждый элемент таблицы это значение индекса строки, возведенное в степень равное индексу столбца

**Для примера:**

| **Результат** |
| --- |
| 1 1 1 1 1  2 4 8 16 32  3 9 27 81 243  4 16 64 256 1024  5 25 125 625 3125 |

for i in range(1, 5 + 1): # строка

for j in range(1, 5 + 1): # столбец

print(i\*\*j, end = ' ')

print()

ыведите на экран таблицу деления числа a на число b, где a и b - числа от 1 до 5, т.е. каждый элемент таблицы это значение частного от деления индекса строки на индекс столбца с точностью 2 знака после запятой. Для форматирования вывода дробного числа используйте функцию round(x, 2) Эта функция принимает на вход число x и количество знаков после запятой, которые нужно оставить при выводе. В данном случае это 2.

**Для примера:**

| **Результат** |
| --- |
| 1.0 0.5 0.33 0.25 0.2  2.0 1.0 0.67 0.5 0.4  3.0 1.5 1.0 0.75 0.6  4.0 2.0 1.33 1.0 0.8  5.0 2.5 1.67 1.25 1.0 |

for i in range(1, 5 + 1): # строка

for j in range(1, 5 + 1): # столбец

print(round(i/j, 2) , end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, элементы каждой строки равны индексу строки

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  0 0 0 0  1 1 1 1  2 2 2 2  3 3 3 3 |

n=int(input())

for i in range( n ): # строка

for j in range(n): # столбец

print(i , end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, заполненную по следующему правилу: элементы четных строк - нули, а нечетных - единицы.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  0 0 0 0  1 1 1 1  0 0 0 0  1 1 1 1 |

n=int(input())

for i in range( n ): # строка

for j in range(n): # столбец

if i %2==0:

print(0 , end = ' ')

else:

print(1 , end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, заполненную по следующему правилу: элементы четных столбцов - нули, а нечетных - единицы.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  0 1 0 1  0 1 0 1  0 1 0 1  0 1 0 1 |

n=int(input())

for i in range( n ): # строка

for j in range(n): # столбец

if j %2==0:

print(0 , end = ' ')

else:

print(1 , end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, заполненную по следующему правилу: строка с индексом 0 и N-1 заполнена 1, все остальные строки 0

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 1 1 1  0 0 0 0  0 0 0 0  1 1 1 1 |

n=int(input())

for i in range( n ): # строка

for j in range(n): # столбец

if i == 0 or i == n-1:

print(1 , end = ' ')

else:

print(0 , end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, заполненную по следующему правилу: ко краям таблицы стоят 1, а в середине 0.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 1 1 1  1 0 0 1  1 0 0 1  1 1 1 1 |

n=int(input())

for i in range( n ): # строка

for j in range(n): # столбец

if i == 0 or i == n-1 or j==0 or j==n-1:

print(1 , end = ' ')

else:

print(0 , end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, по углам которой стоят 1, все остальные элементы 0.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 0 0 1  0 0 0 0  0 0 0 0  1 0 0 1 |

n=int(input())

for i in range(n):

for j in range(n):

if (i==0 and j==0) or (j==n-1 and i==0) or (i==n-1 and j==0) or (j==n-1 and i==n-1):

print(1, end = ' ')

else :

print(0,end = ' ')

print()

Программа получает на вход натуральное число N (2 ≤ N ≤100, N - четное). Выведите на экран таблицу размером NxN, верхняя половина которой заполнена 1, а нижняя 0.

Примечание.

Давайте посмотрим какие строки будут заполнены единицами для данного примера. Это строки 0, 1, а строки 2 и 3 уже нулями. То есть если значение строки i меньше половины N, то нужно заполнять 1, иначе 0. Т.к. N четное, то всегда можно разделить таблицу на 2 половины.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 1 1 1  1 1 1 1  0 0 0 0  0 0 0 0 |

n = int(input())

a = [['0'] \* n for i in range(n)]

for i in range(n):

for j in range (n):

a[n//4][j] = '1'

a[n//6][j] = '1'

for l in a:

print(\*l)

Программа получает на вход натуральное число N (2 ≤ N ≤100, N - четное). Выведите на экран таблицу размером NxN, верхняя четверть заполнена 1, а все остальные элементы 0. Пример

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 1 0 0  1 1 0 0  0 0 0 0  0 0 0 0 |

n = int(input())

a = [['0'] \* n for i in range(n)]

for i in range(n):

for j in range (n):

a[n//4][j//2] = '1'

a[n//6][j//2] = '1'

for l in a:

print(\*l)

Программа получает на вход натуральное число N (3 ≤ N ≤100, N - нечетное). Выведите на экран таблицу размером NxN, средний столбец и средняя строка которой заполнены единицами, все остальные элементы нули.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 5 | 5  0 0 1 0 0  0 0 1 0 0  1 1 1 1 1  0 0 1 0 0  0 0 1 0 0 |

n = int(input())

a = [['0'] \* n for i in range(n)]

for i in range(n):

for j in range (n):

a[i][n//2] = '1'

a[n//2][j] = '1'

for l in a:

print(\*l)

или

n=int(input())

for i in range(n):

for j in range(n):

if i==n//2 or j== n//2:

print(1, end = ' ')

else :

print(0,end = ' ')

print()

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, в которой на главной диагонали стоят 1, а все остальные места заполнен 0.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 0 0 0  0 1 0 0  0 0 1 0  0 0 0 1 |

n=int(input())

a = [[0] \* n for i in range(n)]

for i in range(n):

for j in range(n):

if i < j:

a[i][j] = '0'

elif i > j:

a[i][j] = '0'

else:

a[i][j] = '1'

for row in a:

print(' '.join([str(elem) for elem in row]))

Программа получает на вход натуральное число N (1 ≤ N ≤100). Выведите на экран таблицу размером NxN, в которой на главной диагонали и ниже стоят 1, а все остальные места заполнен 0.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 4 | 4  1 0 0 0  1 1 0 0  1 1 1 0  1 1 1 1 |

n=int(input())

a = [[0] \* n for i in range(n)]

for i in range(n):

for j in range(n):

if i < j:

a[i][j] = 0

else:

a[i][j] = 1

for row in a:

print(' '.join([str(elem) for elem in row]))

Программа получает на вход натуральное число N (2 ≤ N ≤100, N - четное). Выведите на экран таблицу размером NxN, нижний угол первой четверти которой заполнен единицами, а все остальные элементы нулями.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| 6 | 6  1 0 0 0 0 0  1 1 0 0 0 0  1 1 1 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0 |

n=int(input())

for i in range(n):

for j in range(n):

if i>=j and i<n//2:

print(1, end = ' ')

else :

print(0,end = ' ')

print()

На вход программе подается натуральное число. Найдите количество цифр не больше 6 в числе.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 45 | 45 | 45  2 |

n=int(input())

k = 0

while n !=0:

if n%10 <= 6:

k+=1 # k=k+1

n//=10 # n=n//10

print(k)

На вход программе подается 2 натуральных числа. Определите у какого числа сумма цифр больше. В ответе выведите 1, если сумма цифр 1 числа больше, 2, если второго, 0 если сума цифр одинаковая

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 45  90 | 45  90 | 45  90  0 |

s = int(input())

sum = 0

while (s != 0):

sum = sum + s % 10

s = s // 10

s1= int(input())

sum1 = 0

while (s1 != 0):

sum1 = sum1 + s1 % 10

s1 = s1 // 10

if sum==sum1:

print(0)

elif sum>sum1:

print(1)

else:

print(2)

На вход программе подается натуральное число. Найдите самую маленькую цифру числа. При решении этой задачи повторите тему поиска минимума.

a=input()

m = min([int(i) for i in str(a)])

print(m)

На вход программе подается натуральное число. Напишите программу, которая выводит первую цифру числа.

**Для примера:**

| **Тест** | **Ввод** | **Результат** |
| --- | --- | --- |
| 45 | 45 | 45  4 |

n = int(input())

while n//10 !=0:

n//=10 # n=n//10

print(n)

*Напишите программу, которая выводит первое слово переданной её символьной строки. Слово – это последовательность непробельных символов, ограниченная пробелами или концами строки.*

*Входные данные*

 Входная строка содержит произвольную последовательность символов.

*Выходные данные*

 Программа должна вывести второе слово этой строки.

**Для примера:**

| **Ввод** | **Результат** |
| --- | --- |
| Vasya is a pupil. | Vasya is a pupil.  is |