

# Python. Задачи - решебник

---

## Оглавление

Программный тест. Python: ввод и вывод .....	2
Тест-задание 4. Ввод-вывод .....	7
Программный тест по теме 11. Вложенные списки .....	13
Тест-задание 11. Вложенные списки .....	15
Программный тест. Условная инструкция if.....	18
Тест-задание 3. If условия.....	23
Программный тест. Циклы .....	27
Тест-задание 5. Циклы .....	30
Программный тест. Строки .....	37
Тест-задание 6. Строки .....	43
Программный тест. Множество .....	47
Тест-задание 7. Множество.....	48
Программный тест. Списки и кортежи .....	51
Тест-задание 8. Списки и кортежи .....	58
Программный тест 12. Словари .....	63
Тест-задание 12. Словари .....	65
Программный тест 10. Методы списков и строк .....	67
Тест-задание 10. Методы списков и строк .....	73
Программный тест 13. Функции .....	79
Тест-задание 13. Функции .....	80
Программный тест 14. Лямбда-функции .....	85
Тест-задание 14. Лямбда-функции .....	87
Программный тест 9. Модули и пакеты .....	89
Тест-задание 9. Методы и пакеты.....	91
Задание. Модули и пакеты.....	93
Программный тест по теме. Регулярные выражения .....	98
Тест-задание. Регулярные выражения .....	99
Программный тест по теме Объектно-ориентированное программирование .....	100
Тест-задание. Объектно-ориентированное программирование .....	102

## Программный тест. Python: ввод и вывод

**Задание 1.** Напишите программу Python для ввода двух целых чисел в одну строку. И вывода произведения этих чисел

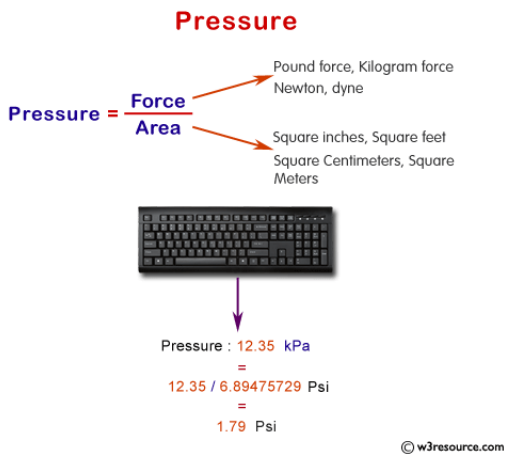
Входные числа: 2 5

Результат: 10

Решение:

```
print("Введите значения x и y")
x, y = map(int, input().split())
print("Произведение x и y равно", x*y)
```

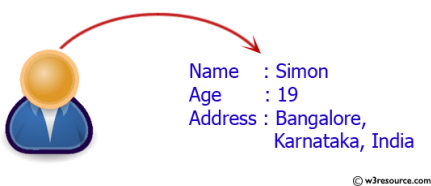
**Задание 2.** Напишите программу на Python для преобразования давления в кПа



Решение:

```
kpa = float(input("Input pressure in in kilopascals> "))
psi = kpa / 6.89475729
print("The pressure in pounds per square inch: %.2f psi" % (psi))
```

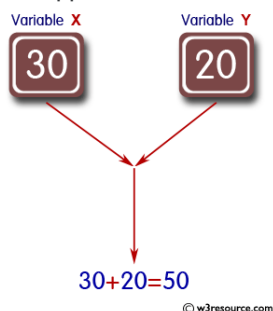
**Задание 3.** Напишите программу Python, которая будет отображать ваши данные, такие как имя, возраст, адрес, в трех разных строках.



Решение:

```
name, age = "Simon", 19
address = "Bangalore, Karnataka, India"
print("Name: {}\nAge: {}\nAddress: {}".format(name, age, address))
```

**Задание 4.** Учитывая переменные  $x = 30$  и  $y = 20$ , напишите программу Python, которая выводит «30 + 20 = 50»



Решение:

```
x = 30
y = 20
print("\n%d+%d=%d" % (x, y, x+y))
```

**Задание 5.** Напишите программу Python для форматирования указанной строки, чтобы ограничить количество символов до 6.

Входные данные: 123456789

Результат: 123456

Решение:

```
str_num = "1234567890"
print()
print('%0.6s' % str_num)
print()
```

**Задание 6.** Напишите программу Python для печати переменной без пробелов между значениями.

Пример значения:  $x = 30$ .

Ожидаемый результат: значение  $x$  равно «30»

Решение:

```
x = 30
print('Value of x is "{}".format(x))
```

**Задание 7.** Напишите программу Python для преобразования целого числа в двоичное, сохраняя ведущие нули.

Пример данных:  $x = 12$

Ожидаемый результат: 00001100

0000001100

Решение:

```
x = 12
print(format(x, '08b'))
print(format(x, '010b'))
```

**Задание 8.** Напишите программу на Python для преобразования десятичного числа в шестнадцатеричное.

### Decimal to Hexadecimal

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Decimal Value = 775

Integer Part	Quotient	Remainder	Remainder in Hexadecimal
775 / 16	48	7	7
48 / 16	3	0	0
3 / 16	0	3	3

Hexadecimal Value = 307

$$(775)_{10} = (307)_{16}$$

Decimal Value = 1256

Integer Part	Quotient	Remainder	Remainder in Hexadecimal
1256 / 16	78	8	8
78 / 16	4	14	E
4 / 16	0	4	4

Hexadecimal Value = 4E8

$$(1256)_{10} = (4E8)_{16}$$

© w3resource.com

Пример десятичного числа: 30

Результат: 1e

Решение:

```
x = 30
print(format(x, '02x'))
x = 4
print(format(x, '02x'))
```

**Задание 9.** Напишите программу, которая находит сумму, произведение и среднее арифметическое трёх целых чисел, введённых с клавиатуры.

*Входные данные*

Три целых числа вводятся в одной строке через пробелы.

*Выходные данные*

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

*Примеры*

входные данные

4 5 7

выходные данные

4+5+7=16

4\*5\*7=140

(4+5+7)/3=5.333

Решение:

```
x,y,z=map(int,input().split())
print("%d+%d+%d=%d" % (x, y, z, (x+y+z)))
print("%d*%d*%d=%d" % (x, y, z, (x*y*z)))
print("(%d+%d+%d)/3=%.3f" % (x, y, z, (x+y+z)/3))
```

**Задание 10.** Напишите программу, которая вводит радиус круга и вычисляет его площадь и длину окружности.

*Входные данные*

Входная строка содержит одно число - радиус круга.

*Выходные данные*

В первой строке должно быть выведена площадь круга, а во второй строке - длина окружности. Оба значения выводятся с тремя знаками в дробной части.

*Примеры*

входные данные

5

выходные данные

78.540

31.416

Решение:

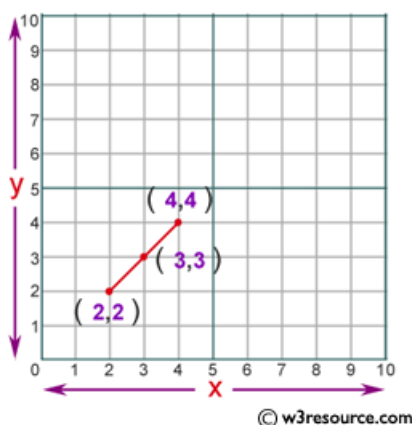
```
from math import pi # Подключим математический пакет для работы с числом пи
r=int(input())
S = pi*r*r
print('%.3f' %S)
l=2*pi*r
print("{:.3f}".format(l))
```

**Задача 11.** Напишите программу на Python для вычисления середины отрезка.

### The Midpoint Formula

$$\begin{matrix} (x_1, y_1) & (x_2, y_2) \\ (2, 2) & (4, 4) \end{matrix}$$

$$\begin{aligned} \text{Midpoint} &= \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \\ &= \left( \frac{2 + 4}{2}, \frac{2 + 4}{2} \right) \\ &= (3, 3) \end{aligned}$$



**Решение**

```
print('\nCalculate the midpoint of a line :')
x1 = float(input('The value of x (the first endpoint) '))
y1 = float(input('The value of y (the first endpoint) '))
x2 = float(input('The value of x (the first endpoint) '))
```

```

y2 = float(input('The value of y (the first endpoint) '))
x_m_point = (x1 + x2)/2

y_m_point = (y1 + y2)/2
print();
print("The midpoint of line is :")
print( "The midpoint's x value is: ",x_m_point)
print( "The midpoint's y value is: ",y_m_point)
print();

```

**Задание 12.** Напишите программу Python для решения  $(x + y) * (x + y)$ .

Тестовые данные:  $x = 4, y = 3$

Ожидаемый результат:  $(4 + 3) ^ 2 = 49$

Решение

```

x, y = 4, 3
result = x * x + 2 * x * y + y * y
print("{} + {} ^ 2) = {}".format(x, y, result))

```

## Тест-задание 4. Ввод-вывод

### Вопрос 1

Выполните 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)
```

### Вопрос 2

Выполните следующие операторы, которые помогут вам разобраться с работой вводом данных в Python

**Чтобы система приняла необходимо убрать пробелы между строками**

```
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
""")
```

### Вопрос 3

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)
```

### Вопрос 4

Выполните следующие операторы, которые помогут вам разобраться с работой format

```
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: {yy1}.".format(xx=2, yy1=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))
```

### **Вопрос 5**

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.

**For example:**

Test	Input	Result
2.0	2.0	2.0 1024.00 0

```
a=float(input())
a=a**10
print('%.3f' %a)
```

### **Вопрос 6**

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.

**For example:**



Test	Input	Result
123	123	123 1 2 3

```
s = input()
print(' '.join(s))
```

### **Вопрос 7**

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

Входные данные

В первой строке вводятся через пробел координаты первой точки сначала x -координата, потом вторая строка y -координата, в третьей и в четвертой строках 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.

**For example:**

Test	Input	Result
1	1	1
1	1	1
2	2	2
2	2	2
		1.41
		4

```
from math import sqrt
x1 = float(input())
y1=float(input())
x2=float(input())
y2=float(input())
l = sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))
print('%.3f' %l)
```

### **Вопрос 8**

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

Входные данные

Три целых числа вводятся с новой строки.

Выходные данные

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

**For example:**

Test	Input	Result
4	4	4
5	5	5
7	7	7
		4+5+7=16
		4*5*7=140
		(4+5+7)/
		3=5.333

```
a = int(input())
b = int(input())
c = int(input())
print(a, '+', b, '+', c, '=', a+b+c, sep = '')
```

```
print(a, '*', b, '*', c, '=', a*b*c, sep = '')  
s1=float((a+b+c)/3)  
aprint('( ', a, '+', b, '+', c, ')/3=', '%.3f' %s1, sep = '')
```

**Вопрос 9**

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

```
a='Python'  
print(a*100)
```

## Программный тест по теме 11. Вложенные списки

### Считывание и вывод двумерных списков

#### Примеры считывания двумерных списков с экрана

**Пример №1** Считать список из **списка целых чисел**

```
n = int(input())
a = []
for i in range(n):
    b = list(map(int, input().split())) #каждый элемент списка a это список b
    a.append(b)
```

**Пример №2** Считать список из **списка строк**

```
n = int(input())
a = []
for i in range(n):
    b = input().split() #каждый элемент списка a это список b
    a.append(b)
```

**Пример №3** Считать список, каждый элемент которого это **список из строки (x) и числа (y)**

```
n = int(input())
a = []
for i in range(n):
    x, y = input().split()
    a.append([x, int(y)])
```

**Пример №4**

```
for i in range(n):
    a.append(list(map(int, input().split())))
```

#### Примеры вывода двумерных списков на экран

```
for x in a:
    print(*x)
for w in a:
    for h in w:
        print(h, end = ' ')
    print()
for x, y in a: #если элемент списка состоит из 2-х значений
    print(x, y)
```

**Пример 1.** Дано нечетное число **n**. Создайте двумерный массив из **n×n** элементов, заполнив его символами "." (каждый элемент массива является строкой из одного символа). Затем заполните символами "\*" среднюю строку массива, средний столбец массива, главную диагональ и побочную диагональ. В результате единицы в массиве должны образовывать

изображение звездочки. Выведите полученный массив на экран, разделяя элементы массива пробелами.

```
n = int(input())
a = [['.' * n for i in range(n)]
for i in range(n):
    a[i][i] = '*'
    a[n // 2][i] = '*'
    a[i][n // 2] = '*'
    a[i][n - i - 1] = '*'
for row in a:
    print(' '.join(row))
```

## Тест-задание 11. Вложенные списки

### Вопрос 1

Найдите индексы первого вхождения максимального элемента.

#### Входные данные

Программа получает на вход размеры массива  $n$  и  $m$ , затем  $n$  строк по  $m$  целых чисел в каждой.

#### Выходные данные

Выведите два числа: номер строки и номер столбца, в которых стоит наибольший элемент в двумерном массиве. Если таких элементов несколько, то выводится тот, у которого меньше номер строки, а если номера строк равны то тот, у которого меньше номер столбца.

**For example:**

Input	Result
3 4	3 4
0 3 2	0 3 2
4	4
2 3 5	2 3 5
5	5
5 1 2	5 1 2
3	3
	1 2

**Answer:(penalty regime: 10, 20, ... %)**

```
n, m = [int(i) for i in input().split()]
a = [[int(j) for j in input().split()] for i in range(n)]
best_i, best_j = 0, 0
curr_max = a[0][0]
for i in range(n):
    for j in range(m):
        if a[i][j] > curr_max:
            curr_max = a[i][j]
            best_i, best_j = i, j
print(best_i, best_j)
```

--

Еще одно решение:

```
n, m= map(int,input().split())
a= []*n
x,y,z= -10**11, -10**11, -10**11
for i in range(n):
    a.append([int(j) for j in input().split()])
for i in range(n):
    for j in range(m):
        if a[i][j] > z:
```

```

        z= a[i][j]
        x= i
        y= j

    print(x,y)

```

## Вопрос 2

1. Считайте с клавиатуры и выведите на экран список из n элементов, каждый элемент – два целых числа. Выведите полученный список на экран.

**For example:**

Input	Result
2	2
78	78 900
900	34 56
34 56	[[78, 900], [34, 56]]

**Answer:(penalty regime: 10, 20, ... %)**

```

a=[]
n=int(input())
for i in range(n):
    a.append(list(map(int,input().split())))
print(a)

```

## Вопрос 3

Проверьте, является ли двумерный массив симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний.

**For example:**

Input	Result
3	3
0 1	0 1
2	2
1 5	1 5
3	3
2 3	2 3
4	4
	yes
3	3
0 0	0 0
0	0
0 0	0 0
0	0
1 0	1 0
0	0
	no

**Answer:(penalty regime: 10, 20, ... %) (ОБА ПРАВИЛЬНЫЕ)**



```

n = int(input())
a = []
for i in range(n):
    row = input().split()
    for j in range(n):
        row[j] = int(row[j])
    a.append(row)
m = 0
for i in range(1, n):
    k = 0
    for j in range(i):
        if (a[i][j] == a[j][i]):
            k = k+1;
    if k == i:
        m = m+1
if (m == n-1):
    print('yes')
else:
    print('no')

```

```

n = int(input())
arr = []
for i in range(n):
    arr.append(list(map(int,
input().split()))))

flag = "yes"

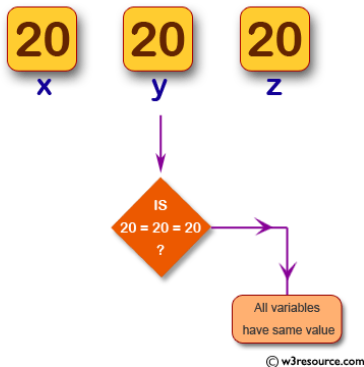
for i in range(len(arr)):
    for j in range(len(arr[i])):
        if arr[i][j] != arr[j][i]:
            flag = "no"
            break
print(flag)

```

## Программный тест. Условная инструкция if

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1) Напишите программу Python, чтобы проверить, имеют ли несколько переменных одно и то же значение.



Решение:

```
x = 20
y = 20
z = 20
if x == y == z == 20:
    print("All variables have same value!")
```

2) Напишите программу Python для выполнения действия, если условие выполняется.

Для данного имени переменной, если значение равно 1, отобразить строку «Первый день месяца!» и ничего не делать, если значение не равно.

Решение:

```
n=1
if n == 1:
    print("\nПервый день месяца!")
    print()
```

3) Напишите программу Python, чтобы определить, является ли данное число (принимаемое пользователем) четным или нечетным, распечатайте соответствующее сообщение для пользователя.

Четные числа от 1 до 100:	Нечетные числа от 1 до 100:
---------------------------	-----------------------------

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

© w3resource.com

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

© w3resource.com

Решение:

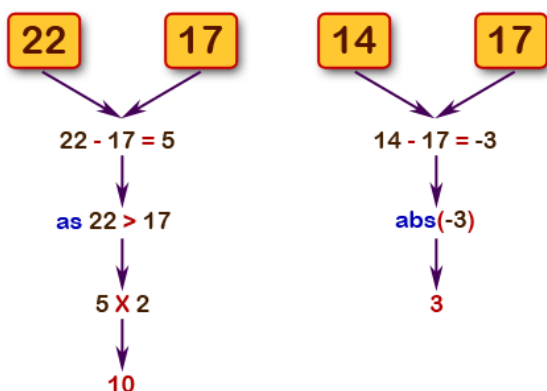
```
num = int(input("Enter a number: "))
mod = num % 2
if mod > 0:
    print("This is an odd number.")
else:
    print("This is an even number.")
```

4) Напишите программу на Python, чтобы проверить, является ли число положительным, отрицательным или нулевым.

Решение

```
num = float(input("Input a number: "))
if num > 0:
    print("It is positive number")
elif num == 0:
    print("It is Zero")
else:
    print("It is a negative number")
```

5) Напишите программу на Python, чтобы получить разницу между заданным числом и 17, если число больше 17, верните удвоенную абсолютную разницу.

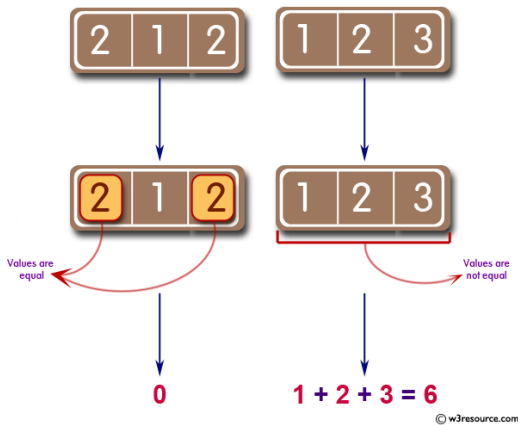


© w3resource.com

Решение:

```
n=int(input())
if n <= 17:
    print(17 - n)
else:
    print((n - 17) * 2)
```

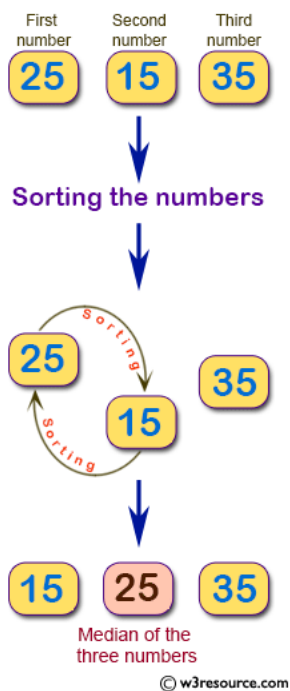
6) Напишите программу Python для суммы трех заданных целых чисел. Однако, если два значения равны, сумма будет равна нулю.



Решение

```
x,y,z=map(int,input().split())
if x == y or y == z or x==z:
    print(x,y,z)
else:
    print(x + y + z)
```

7) Напишите программу на Python, чтобы найти медиану среди трех заданных чисел.



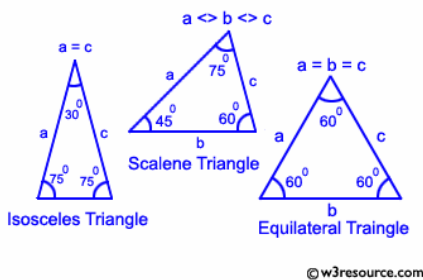
Решение:

```

x = input("Input the first number")
y = input("Input the second number")
z = input("Input the third number")
print("Median of the above three numbers -")
if y < x and x < z:
    print(x)
elif z < x and x < y:
    print(x)
elif z < y and y < x:
    print(y)
elif x < y and y < z:
    print(y)
elif y < z and z < x:
    print(z)
elif x < z and z < y:
    print(z)

```

8) Напишите программу на Python, чтобы проверить, образуют ли три заданные длины (целые числа) трех сторон прямоугольный треугольник. Выведите «Да», если заданные стороны образуют прямоугольный треугольник, в противном случае выведите «Нет».



Решение

```

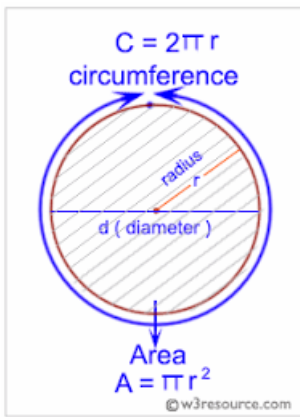
print("Input three integers(sides of a triangle)")
int_num = list(map(int,input().split()))
x,y,z = sorted(int_num)
if x**2+y**2==z**2:
    print('Yes')
else:
    print('No')

```

9) Есть две окружности C1 с радиусом r1, центральной координатой (x1, y1) и C2 с радиусом r2 и центральной координатой (x2, y2).

Напишите программу Python, чтобы проверить следующее:

- "C2 находится в C1", если C2 находится в C1
- "C1 находится в C2", если C1 находится в C2
- «Окружности C1 и C2 пересекаются», если окружности C1 и C2 пересекаются, и
- «C1 и C2 не пересекаются», если C1 и C2 не перекрываются.



Решение

```
import math
print("Input x1, y1, r1, x2, y2, r2:")
x1,y1,r1,x2,y2,r2 =map(float, input().split())
d = math.sqrt((x1-x2)**2 + (y1-y2)**2)
if d < r1-r2:
    print("C2 is in C1")
elif d < r2-r1:
    print("C1 is in C2")
elif d > r1+r2:
    print("Circumference of C1 and C2 intersect")
else:
    print("C1 and C2 do not overlap")
```

### Тест-задание 3. If условия

#### Вопрос 1

Найдите максимум из двух чисел.

Входные данные

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

Выходные данные

Выведите наибольшее из данных чисел.

**For example:**

Test	Input	Result
1	1	1
2	2	2
		2

```
num1 = int(input())
num2 = int(input())
if num1 > num2:
    print(num1)
else:
    print(num2)
```

#### Вопрос 2

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.

**For example:**

Test	Input	Result
200	2007	2007
7		NO
200	2000	2000
0		YES

```
num1 = int(input())
if ((num1 % 400 == 0) or ((num1 % 4 == 0) and (num1 % 100 != 0))):
    print('YES')
else:
    print('NO')
```

#### Вопрос 3

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`

**For example:**

Input	Result
1534	1534 1
-42	-42 -1
0	0 0

```
num1 = int(input())
if (num1 < 0):
    print(-1)
elif (num1 > 0):
    print(1)
else: print(0)
```

#### **Вопрос 4**

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.

**For example:**

Test	Input	Result
1	1	1
2	2	2
3	3	3
		0

```
num1 = int(input())
num2 = int(input())
num3 = int(input())
if (num1 == num2) and (num1 == num3) and (num2 == num3):
    print(3)
elif (num1 == num2) or (num1 == num3) or (num2 == num3):
    print(2)
else: print(0)
```

#### **Вопрос 5**

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 введен неверный номер месяца.

**For example:**

Test	Input	Result
4	4	4 30



**Answer:(penalty regime: 10, 20, ... %)**

```
num1 = int(input())
if (num1 == 2):
    print(28)
elif (num1 == 1) or (num1 == 3) or (num1 == 5) or (num1 == 7) or (num1
== 8) or (num1 == 10) or (num1 == 12):
    print(31)
elif (num1 == 4) or (num1 == 6) or (num1 == 9) or (num1 == 11):
    print(30)
else: print(0)
```

### **Вопрос 6**

В условии одной из задач на этом сайте написано: "Если данное четырехзначное число является симметричным, выведите 1, иначе выведите **любое другое** целое число". Для проверки задачи используются заранее подготовленные примеры и правильные ответ на них

Школьнику кажется, что он решил эту задачу, но тестирующая система почему-то не принимает его решение. Школьник думает, что это происходит оттого, что он выводит не то **любое другое** число, которое записано в правильных ответах.

Напишите программу, которая по ответу, записанному в тестирующей системе и по ответу школьника определяет, верно ли школьник решил задачу.

Входные данные

Программа получает на вход два числа: ответ, записанный в тестирующей системе и ответ школьника.

Выходные данные

Программа должна вывести YES, если школьник дал верный ответ и NO в противном случае.

**For example:**

Test	Input	Result
11	11	11
-1	-1	-1
		YES
3	3	3
1	1	1
		NO

```
sc = int(input())
sys = int(input())
if (sc == 1) and (sys == 1):
    print("YES")
elif (sys != 1) and (sc != 1):
    print("YES")
else: print("NO")
```

### **Вопрос 7**

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'.

**For example:**

Test	Input	Result
4	4	4 sprin g
25	25	25 NO

```
num1 = int(input())
if (num1 == 12) or (num1 == 1) or (num1 == 2):
    print('winter')
elif (num1 == 3) or (num1 == 4) or (num1 == 5):
    print('spring')
elif (num1 == 6) or (num1 == 7) or (num1 == 8):
    print('summer')
elif (num1 == 9) or (num1 == 10) or (num1 == 11):
    print('autumn')
else: print('NO')
```

## Программный тест. Циклы

Приведем примеры разобранные в видеолекциях темы. На страницах теста вам выполнить задания на тему "Цикл for", "Цикл while" и "Вложенные циклы".

1) Написать программу Python для печати \* 10 раз в строчку?

Пример вывода:

\*\*\*\*\*

Решение

```
for i in range(10):  
    print('*', end='')
```

2) Напишите программу Python для построения следующего шаблона, используя вложенный цикл for.

**Решение**

```
*  
* *  
* * *  
* * * *  
* * * *  
* * * *  
* * * *  
* * *  
* *  
*  
  
for i in range(5):  
    for j in range(i):  
        print('*', end='')  
    print()  
for i in range(5):  
    for j in range(5-i):  
        print('*', end='')  
    print()
```

3) 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  $N \times N$ , состоящий из знаков '\*'.

**Ввод**    **Вывод**

5

\*\*\*  
\*\*

\*\*\*  
\*\*

5

\*\*\*  
\*\*

\*\*\*  
\*\*

\*\*\*  
\*\*

```
n = int(input())
for i in range(n):
    for j in range(n):
        print('*', end='')
    print()
```

4) Напишите программу на Python для создания таблицы умножения (от 1 до 10) числа.

```
a = 1
while a <= 10:
    b = 1
    while b <= a:
        c = a*b
        print(c, end=" ")
        b += 1
    print(" ")
    a += 1
```

5) Напишите функцию Python, которая принимает положительное целое число и возвращает сумму куба всех положительных целых чисел, меньших указанного числа.

Например:  $8 = 7^3 + 6^3 + 5^3 + 4^3 + 3^3 + 2^3 + 1^3 = 784$

```
n = int(input())
n -= 1
total = 0
while n > 0:
    total += n * n * n
    n -= 1
print("Sum of cubes: ", total)
```

6) Напишите программу Python, которая принимает положительное число и вычитает из этого числа сумму его цифр и т. д. Продолжает эту операцию, пока число не станет положительным.

```
n=int(input())
s = 0
n_str = str(n)
while (n > 0):
    n -= sum([int(i) for i in list(n_str)])
    n_str = str(n)
    s += 1
print(s)
```

7) Напишите программу Python для создания гистограммы из заданного списка целых чисел.

Character to Print → @

Format of Histogram →



```
@ @
@ @ @
@ @ @ @ @ @
@ @ @ @ @
```

© w3resource.com

```
items=list(map(int, input().split() ))
for n in items:
    output = ''
    times = n
    while( times > 0 ):
        output += '*'
        times = times - 1
    print(output)
```

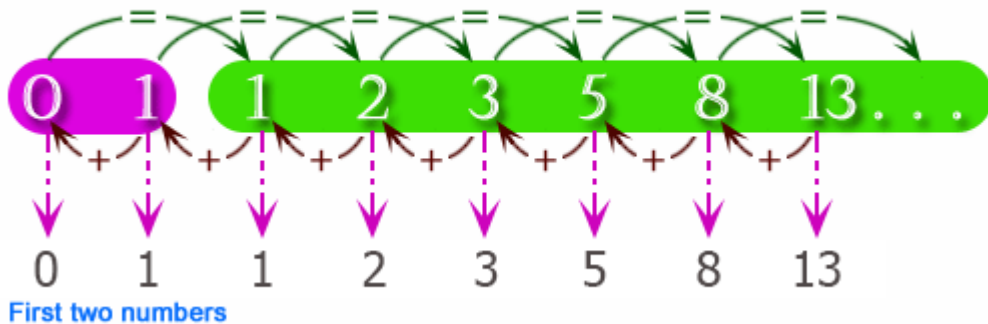
8) Напишите программу на Python, чтобы получить ряд Фибоначчи от 0 до 50.

Примечание. Последовательность Фибоначчи - это последовательность чисел:

0, 1, 1, 2, 3, 5, 8, 13, 21, ....

Каждое следующее число находится путем сложения двух чисел перед ним.

*Fibonacci Sequence :*



© w3resource.com

```
x,y=0,1
while y<50:
    print(y)
    x,y = y,x+y
```

## Тест-задание 5. Циклы

### Вопрос 1

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

**For example:**

Input	Result
1	1
10	10
	1 2 3 4 5 6 7 8 9
	10

**Answer:(penalty regime: 10, 20, ... %)**

```
a = int(input())
b = int(input())
for i in range(a, b+1):
    print(i, end = ' ')
```

### Вопрос 2

Даны два целых числа A и B. 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.

**For example:**

Test	Input	Result
1	1	1
10	10	10
		1 2 3 4 5 6 7 8 9
		10

**Answer:(penalty regime: 10, 20, ... %)**

```
a=int(input())
b=int(input())
if a<b:
    for i in range (a, b+1):
        print(i, end = ' ')
else:
    for i in range (a, b-1, -1):
        print(i, end = ' ')
```

### Вопрос 3

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

**For example:**

Test	Input	Result
7	7	7
1	1	1
		7 5 3
		1

**Answer:(penalty regime: 10, 20, ... %)**

```
a = int(input())
b = int(input())
for i in range(a, b-1, -1):
    if i%2 == 1:
        print(i, end = ' ')
```

#### **Вопрос 4**

##### Входные данные

Вводятся целые числа а и b. Гарантируется, что а не превосходит b

##### Выходные данные

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

**For example:**

Test	Input	Result
2	2	2
5	5	5
		2 4

**Answer:(penalty regime: 10, 20, ... %)**

```
a = int(input())
b = int(input())
for i in range(a, b+1):
    if i%2 == 0:
        print(i, end = ' ')
```

#### **Вопрос 5**

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.

**For example:**

Test	Input	Result
6	6	6
12	12	12
		72

**Answer:(penalty regime: 10, 20, ... %)**

```
a = int(input())
```

```

b = int(input())
sum = 0
for i in range(0, abs(b), 1):
    sum += abs(a)
if ((a > 0 and b > 0) or (a < 0 and b < 0)):
    print(sum)
else:
    print (-sum, sep = '')

```

### **Вопрос 6**

#### 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.

**For example:**

Test	Input	Result
2	2	2
8	8	8
		4

**Answer:(penalty regime: 10, 20, ... %)**

```

a = int(input())
b = int(input())
for i in range(b+1):
    if i*i>=a and i*i<=b:
        print(i*i, end = ' ')

```

или

```

a = int (input())
b = int (input())
for i in range (a, b+1):
    if i**0.5 == round(i**0.5):
        print (i)

```

### **Вопрос 7**

Факториалом числа  $n$  называется произведение  $1 \times 2 \times \dots \times n$ . Обозначение:  $n!$ .



Number

4



The Factorial of 4 is

$$4 \times 3 \times 2 \times 1 = 24$$

© w3resource.com

По данному натуральному  $n$  вычислите значение  $n!$ . Пользоваться математической библиотекой `math` в этой задаче запрещено.

**For example:**

Test	Input	Result
4	4	4 24

**Answer:(penalty regime: 10, 20, ... %)**

```
a = int(input())
sum = a
while a > 1:
    a -= 1;
    sum *= a;

print(sum)
```

### **Вопрос 8**

#### **Входные данные**

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

#### **Выходные данные**

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

**For example:**

Input	Result
4	4
9	9
0	0
8	8
1	1
1	1

**Answer:(penalty regime: 10, 20, ... %)**

```
a = int(input())
i = 0
sum = 0
while i < a:
    i += 1
    b = int(input())
```

```

        if b == 0:
            sum += 1
    print(sum)

```

или

```

num_zeroes = 0
for i in range(int(input())):
    if int(input()) == 0:
        num_zeroes += 1
print(num_zeroes)

```

### **Вопрос 9**

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

**For example:**

Input	Result
32	32 1 2 4 8 16 32

**Answer:(penalty regime: 10, 20, ... %)**

```

a = int(input())
for i in range(1, a + 1):
    if (a % i == 0):
        print (i, end = ' ')

```

### **Вопрос 10**

Напишите программу, которая считает сумму цифр введенного числа.

#### **Входные данные**

Входная строка содержит одно натуральное число.

#### **Выходные данные**

Программа должна вывести сумму цифр введенного числа.

**For example:**

Input	Result
1234	1234
5	5
	15

**Answer:(penalty regime: 10, 20, ... %)**

```

a=int(input())
a_str = str(a)
sum1 = 0
for i in list(a_str):
    sum1 += int(i)

```

```
print(sum1)
```

### **Вопрос 11**

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.

**For example:**

Input	Result
1	1
7	7
9	9
0	0
	17

**Answer:(penalty regime: 10, 20, ... %)**

### **Вопрос 12**

Последовательность состоит из натуральных чисел, не превосходящих 109109, 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.

**For example:**

Input	Result
1	1
7	7
9	9
0	0
	9

**Answer:(penalty regime: 10, 20, ... %)**

### **Вопрос 13**

**For example:**

Input	Result
8	8
	6

**Answer:(penalty regime: 10, 20, ... %)**

#### **Вопрос 14**

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.

**For example:**

Input	Result
1	1
7	7
9	9
0	0
	2

**Answer:(penalty regime: 10, 20, ... %)**

#### **Вопрос 15**

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  $N \times N$  , состоящий из знаков '\*'.

**For example:**

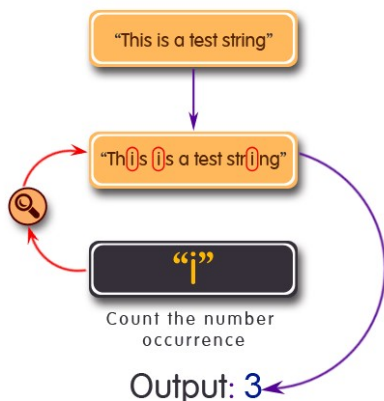
Input	Result
5	5 *

**Answer:(penalty regime: 10, 20, ... %)**

## Программный тест. Строки

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

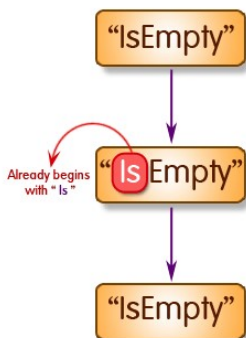
1. Напишите программу Python для подсчета количества вхождений определенного символа в строку.



Решение

```
s = "The quick brown fox jumps over the lazy dog."
print()
print(s.count("q"))
print()
```

2. Напишите программу Python, чтобы получить новую строку из заданной строки, где «Is» было добавлено впереди. Если данная строка уже начинается с «Is», вернуть строку без изменений.



Решение

```
str='IsEmpty'
if len(str) >= 2 and str[:2] == "Is":
    print(str)
else:
    print("Is" + str)
```

3. Напишите программу Python, которая принимает имя файла от пользователя и распечатывает его расширение.

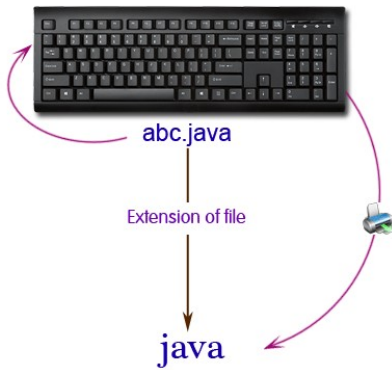
Пример файла: abc.java

Функция Python `str.split (sep = None, maxsplit = -1)`:

Функция возвращает список слов заданной строки с использованием разделителя в качестве строки-разделителя.

- Если задано *maxsplit* , список будет содержать не более *maxsplit* + 1 элементов.
- Если *maxsplit* не указан или -1, то количество разделений не ограничено.

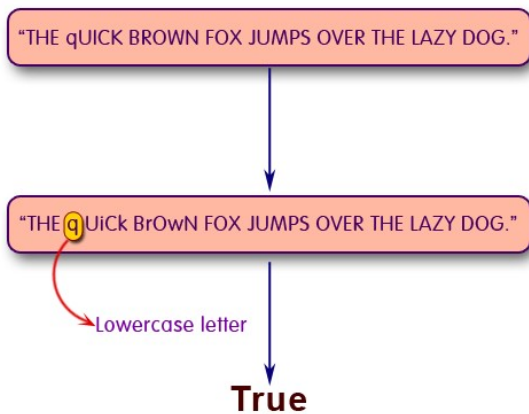
- Если задано *sep*, последовательные разделители не группируются вместе и считаются ограничивающими пустые



Решение

```
filename = input("Input the Filename: ")
f_extns = filename.split(".")
print ("The extension of the file is : " + repr(f_extns[-1]))
```

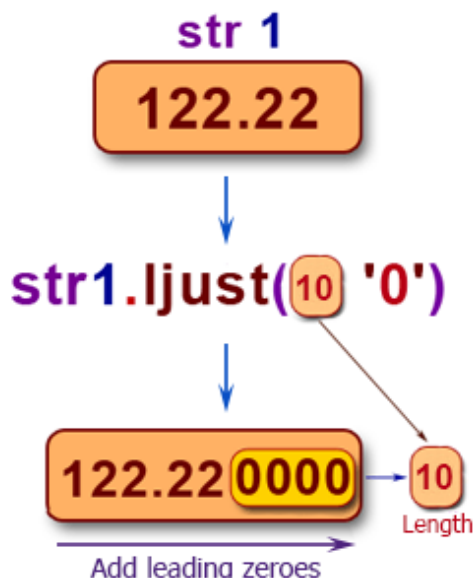
4. Напишите программу на Python, чтобы проверить, существуют ли прописные буквы в строке.



Решение

```
str1 = 'A8238i823acdeOUEI'
print(any(c.islower() for c in str1))
```

5. Напишите программу Python для добавления конечных и ведущих нулей к строке.



## Решение

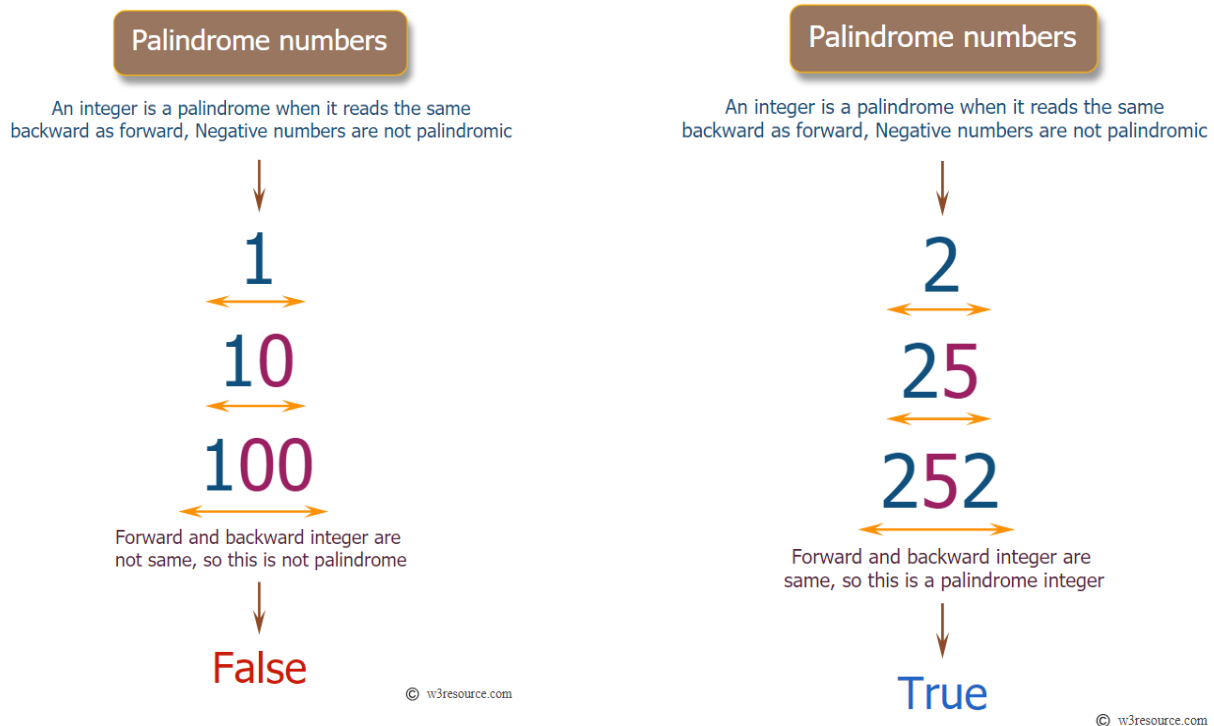
```
str1='122.22'  
print("Original String: ",str1)  
print("\nAdded trailing zeros:")  
str1 = str1.ljust(8, '0')  
print(str1)  
str1 = str1.ljust(10, '0')  
print(str1)  
print("\nAdded leading zeros:")  
str1='122.22'  
str1 = str1.rjust(8, '0')  
print(str1)  
str1 = str1.rjust(10, '0')  
print(str1)
```

6. Напишите программу на Python, чтобы проверить, является ли данное целое число палиндромом.

Примечание. Целое число является палиндромом, когда оно читается так же, как вперед и назад. Отрицательные числа не являются палиндромами.

123, 65699 - не являются палиндромами

121, 4224 – палиндром



## Решение

```
str1='122.221'  
if str1 == str1[::-1]:  
    print("True")  
else:  
    print("False")
```

7. Напишите программу на Python, чтобы перевернуть цифры данного числа и добавить его к исходному. Если сумма не является палиндромом, повторите эту процедуру.

Примечание: палиндром - это слово, число или другая последовательность символов, которая читается так же, как вперед и назад, например, kazak

$$\begin{array}{r}
 1234 \\
 \downarrow \\
 \begin{array}{r}
 1234 \\
 \text{Reverse the digits} \\
 + 4321 \\
 \hline
 5555 \\
 \text{Palindrome number}
 \end{array}
 \end{array}$$

( Palindrome numbers remain the same whether written forwards or backwards )

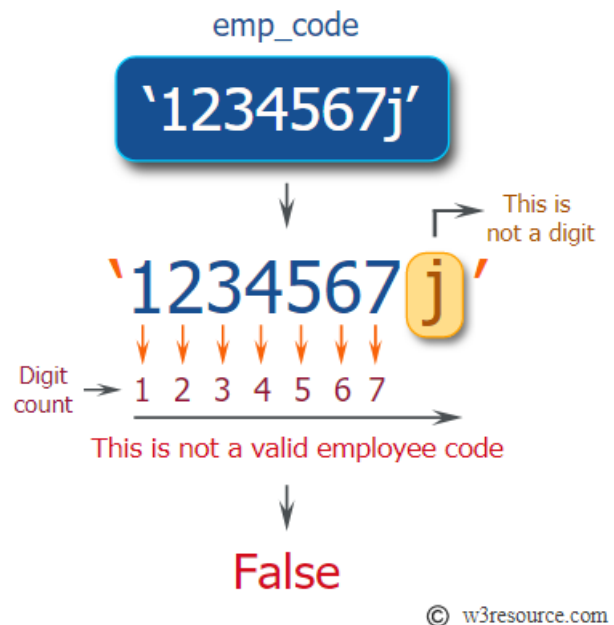
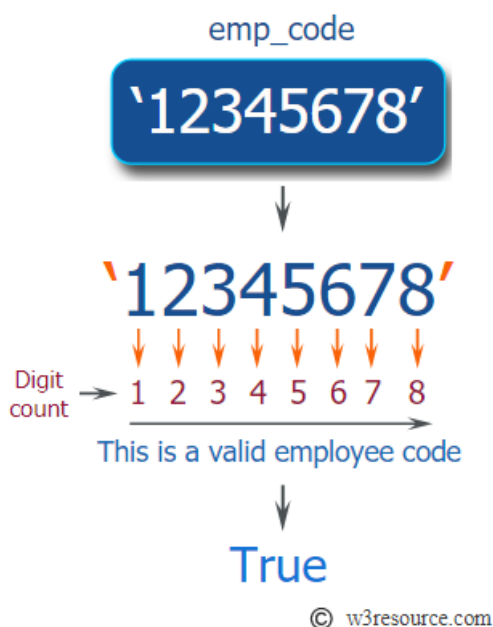
Решение

```

n=1234
s = 0
while True:
    k = str(n)
    if k == k[::-1]:
        break
    else:
        m = int(k[::-1])
        n += m
s += 1
print(n)

```

8. Напишите программу на Python, чтобы проверить, состоит ли данный код сотрудника ровно из 8 или 12 цифр. Верните True, если код сотрудника действителен, и False, если это не так.



Решение

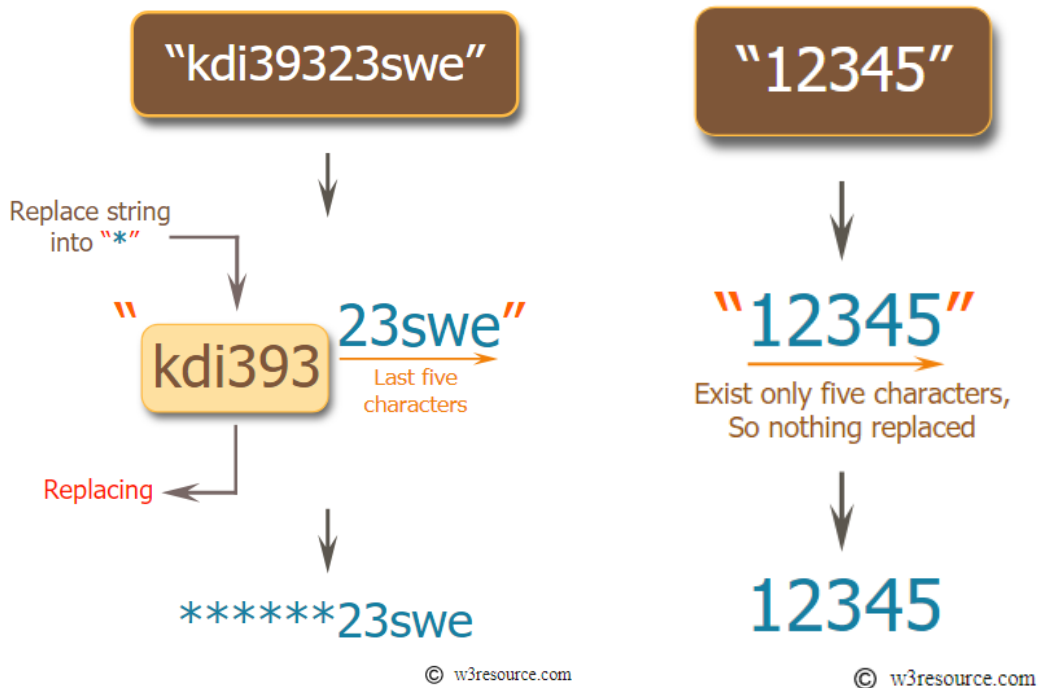
```

n='12345678j'
print(len(n) in [8, 12] and n.isdigit())

```



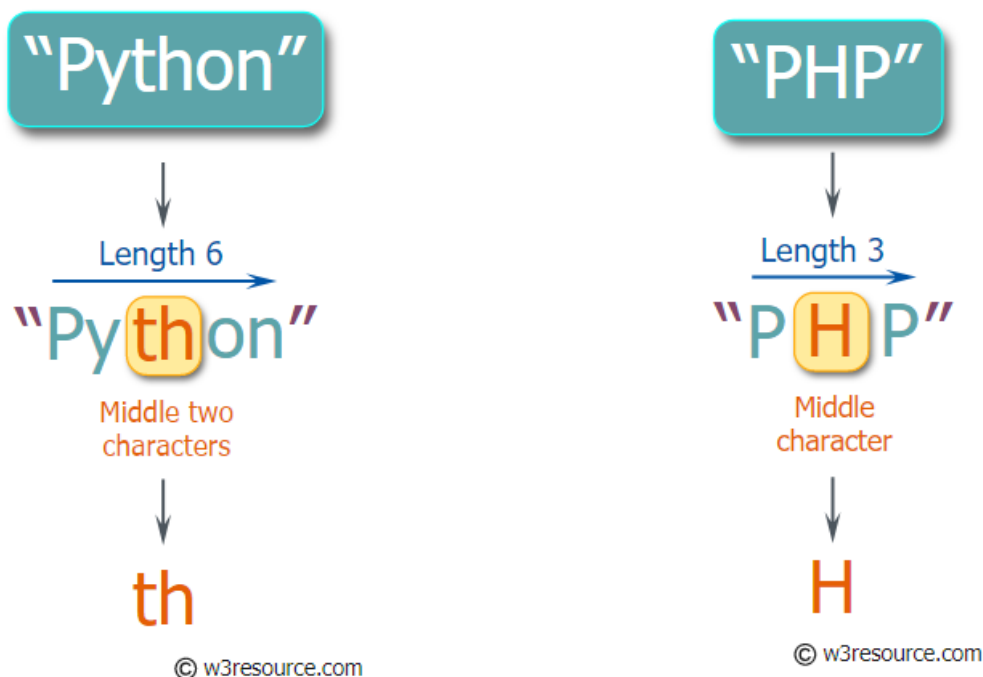
9. Напишите программу Python, которая заменяет все символы данной строки, кроме последних пяти, на «\*» и возвращает новую строку с маской.



Решение

```
n='kdi39323swe'
print(''* (len(n)-5) + n[-5:])
```

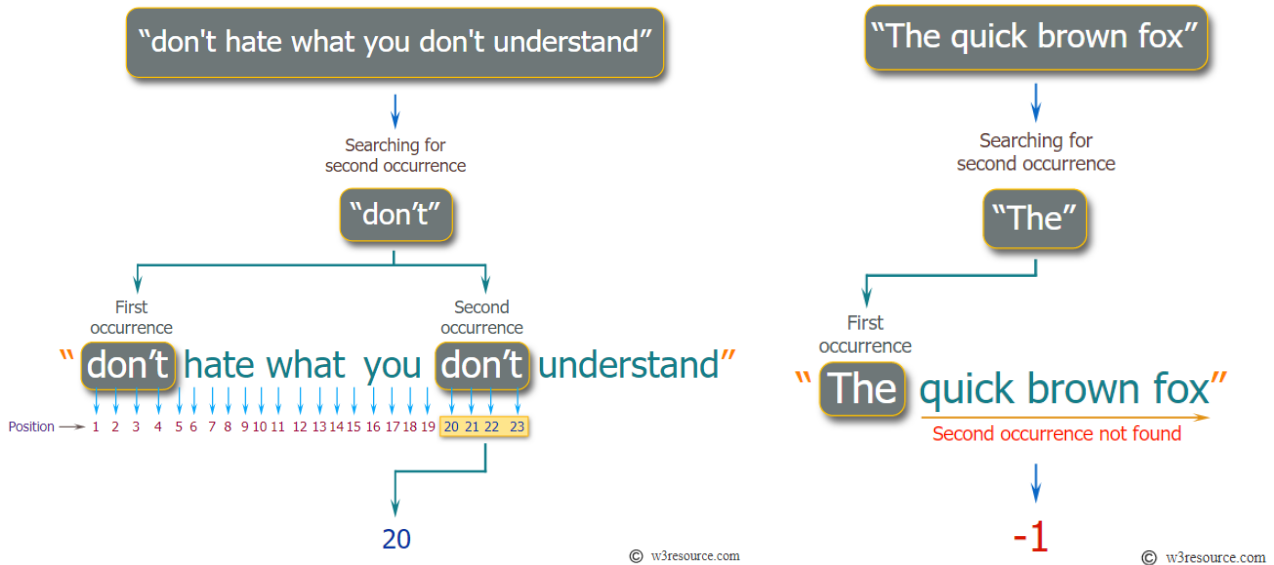
10. Напишите программу Python для поиска среднего символа (ов) данной строки. Если длина строки нечетная, верните средний символ и верните два средних символа, если длина строки четная.



## Решение

```
txt="Python"
print(txt[(len(txt)-1)//2:(len(txt)+2)//2])
```

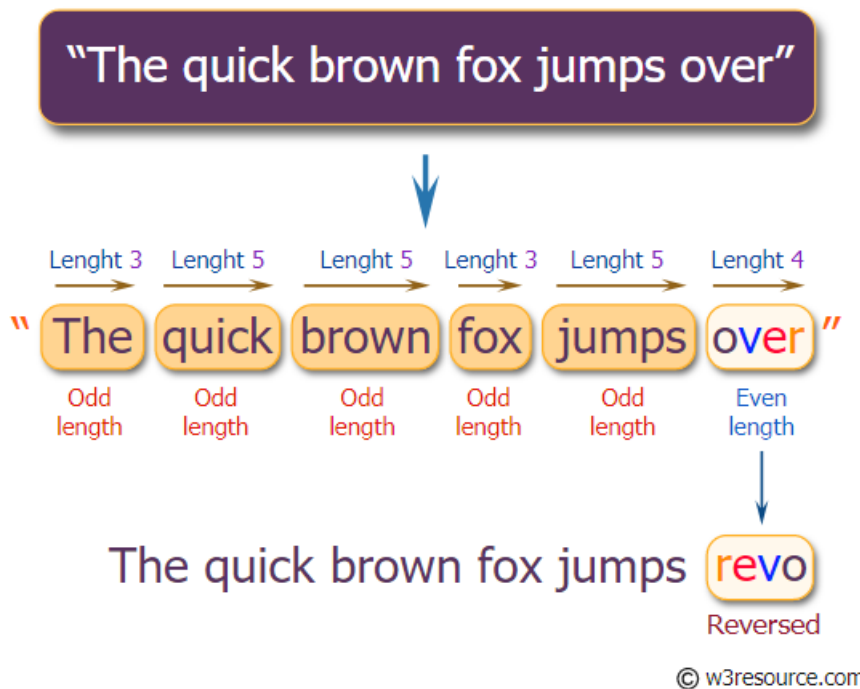
11. Напишите программу Python, чтобы найти позицию второго вхождения данной строки в другой данной строке. Если такой строки нет, верните -1.



## Решение

```
txt="The quick brown fox jumps over the lazy dog"
str1="the"
print(txt.find(str1, txt.find(str1)+1))
```

12. Напишите программу на Python, которая перевернет все слова четной длины.



## Решение

```
txt="The quick brown fox jumps over the lazy dog"
print(' '.join(i[::-1] if not len(i)%2 else i for i in txt.split()))
```

## Тест-задание 6. Строки

### Вопрос 1

На языке программирования Python нельзя использовать циклы.

#### Входные данные

Дана строка.

#### Выходные данные

Сначала выведите третий символ этой строки.

Во второй строке выведите предпоследний символ этой строки.

В третьей строке выведите первые пять символов этой строки.

В четвертой строке выведите всю строку, кроме последних двух символов.

В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого).

В шестой строке выведите все символы с нечетными индексами, то есть начиная со второго символа строки.

В седьмой строке выведите все символы в обратном порядке.

В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.

В девятой строке выведите длину данной строки.

Answer:(penalty regime: 10, 20, ... %)

```
str1 = input()
print(str1[2])
print(str1[-2])
print(str1[:5])
print(str1[:-2])
i = 0
while i < len(str1):
    print(str1[i], end = " ")
    i = i + 2
print("")
j = 1
while j < len(str1):
    print(str1[j], end = " ")
    j = j + 2
print("")
print(str1[-1::-1])
print(str1[-1::-2])
print(len(str1))
```

### Вопрос 2

Напишите программу, которая выводит первое слово переданной ей 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.

**For example:**

Input	Result
Vasya is a pupil.	Vasya is a pupil. is

Answer:(penalty regime: 10, 20, ... %)

```
str1 = input()
```

```
str2 = str1.split(' ')
print(str2[1])
```

### **Вопрос 3**

Дана строка, состоящая из слов, разделенных пробелами. Определите, сколько в ней слов. Используйте для решения задачи метод count.

#### **Входные данные**

Вводится строка.

#### **Выходные данные**

Выведите ответ на задачу.

**For example:**

Input	Result
Hello world	Hello world 2

**Answer:(penalty regime: 10, 20, ... %)**

```
str1 = input()
print(str1.count(' ') + 1)
```

### **Вопрос 4**

Напишите программу, которая определяет, является ли введенное слово (без пробелов) палиндромом, то есть верно ли, что оно читается одинаково слева направо и справа налево.

Входные данные

Входная строка содержит слово – последовательность символов без пробелов.

Выходные данные

Программа должна вывести слово 'YES', если введенное слово – палиндром, и слово 'NO', если оно не является палиндромом.

**For example:**

Input	Result
kaza k	kaza k YES
bana n	bana n NO

**Answer:(penalty regime: 10, 20, ... %)**

```
str1 = input()
if str1 == str1[::-1]:
    print("YES")
else:
    print("NO")
```

### **Вопрос 5**

Дана строка, состоящая ровно из двух слов, разделенных пробелом. Переставьте эти слова местами. Результат запишите в строку и выведите получившуюся строку.

При решении этой задачи нельзя пользоваться циклами и инструкцией if.

#### **Входные данные**

Вводится строка.

Выходные данные

Выведите ответ на задачу.

**For example:**

Input	Result
Hello, world!	Hello, world! world! Hello,

**Answer:(penalty regime: 10, 20, ... %)**

```
str1 = input()
str2 = str1.split(' ')
str3 = str2[1] + ' ' + str2[0]
print(str3)
```

**Вопрос 6**

В операционной системе Windows путь к файлу состоит из имени логического диска и названий каталогов, в которых находится этот файл. Части пути разделены знаками "\" (обратный слэш). Требуется разбить заданный путь к файлу на части — имя диска и названия каталогов.

Входные данные

Входная строка содержит путь к файлу.

Выходные данные

Программа должна вывести имя диска и названия всех каталогов, составляющих путь. Каждый элемент выводится в отдельной строке.

**For example:**

Input	Result
C: \WINDOWS\MEDIA\work.wav	C: \WINDOWS\MEDIA\work.wav C: WINDOWS MEDIA work.wav

**Answer:(penalty regime: 10, 20, ... %)**

```
str1 = input()
str2 = str1.split("\\")
for i in range(len(str2)):
    print(str2[i])
```

**Вопрос 7**

Напишите программу, которая выводит самое длинное 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 в первой строке самое длинное слово переданной ей строки, а во второй – длину этого слова. Если слов максимальной длины несколько — вывести первое встретившееся слово максимальной длины.

**For example:**

Input	Result
abra cadabra fibra	abra cadabra fibra cadabra 7

**Answer:(penalty regime: 10, 20, ... %)**

```
str1 = input()
str2 = str1.split(' ')
best = 0
for index in range(len(str2)):
    if len(str2[index]) > len(str2[best]):
        best = index
print(str2[best])
print(len(str2[best]))
```

### **Вопрос 8**

Напишите программу, которая преобразует строку, содержащую имя, отчество и фамилию человека, к форме

<фамилия> <инициалы>

#### **Входные данные**

Входная строка содержит имя, отчество и фамилию, разделённые одиночными пробелами.

#### **Выходные данные**

Программа должна вывести в одной строке сначала фамилию, а потом (через пробел) – инициалы.

**For example:**

Input	Result
Donald Ervin Knuth	Donald Ervin Knuth Knuth D.E.

**Answer:(penalty regime: 10, 20, ... %)**

```
str1 = input()
str2 = str1.split(' ')
a=str(str2[0])
b=str(str2[1])
print(str2[2], a[0] + '.'+ b[0]+'.')
```

## Программный тест. Множество

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1) Напишите программу Python для удаления элемента (ов) из множества.

Решение

```
num_set = set([0, 1, 3, 4, 5])
num_set.pop()
print(num_set)
num_set.pop()
print(num_set)
```

2) Напишите программу Python для создания пересечения множеств.

Решение

```
setx = set(["green", "blue"])
sety = set(["blue", "yellow"])
seta = setx | sety
print(seta)
```

3) Напишите программу на Python, чтобы найти длину множества.

Решение

```
seta = set([5, 10, 3, 15, 2, 20])
print(len(seta))
```

## Тест-задание 7. Множество

### Вопрос 1

Напишите программу, которая считает знаки пунктуации в символьной строке. К знакам пунктуации относятся символы из набора ".,:;!?".

#### Входные данные

Программа получает на вход символьную строку.

#### Выходные данные

Программа должна вывести общее количество знаков пунктуации во входной строке.

For example:

Input	Result
Hi, guys!	Hi, guys! 2

Answer:(penalty regime: 10, 20, ... %)

```
a = input()
b = {'.', ',', ';', ':', '!', '?'}
numb = 0
for i in a:
    if (i in b):
        numb += 1
print(numb)
```

### Вопрос 2

Напишите программу, которая находит все различные цифры в символьной строке.

#### Входные данные

На вход программе подаётся символьная строка.

#### Выходные данные

Программа должна вывести в одной строке все различные цифры, которые встречаются в исходной строке, в порядке возрастания. Если в строке нет цифр, нужно вывести слово 'NO'.

For example:

Input	Result
ab1n32kz 2	ab1n32kz 2 123

Answer:(penalty regime: 10, 20, ... %)

```
string = input()
numbers = set("0123456789")
str_in = set(string)
c = numbers & str_in
if len(c)>0:
    for i in sorted(c):
        print(i, end="")
else: print("NO")
```



### Вопрос 3

Напишите программу, которая выводит все цифры, встречающиеся в символьной строке больше одного раза.

#### Входные данные

Входная строка может содержать содержит цифры, пробелы и латинские буквы.

#### Выходные данные

Программа должна вывести в одну строчку в порядке возрастания все цифры, встречающиеся во входной строке больше одного раза. Если таких цифр нет, нужно вывести слово 'NO'.

Answer:(penalty regime: 10, 20, ... %)

```
result = set()
s = input()
for i in s:
    if i.isdigit() and -1 < s.find(i) < s.rfind(i):
        result.add(i)
print(''.join(sorted(result)) if len(result) > 0 else 'NO')
```

### Вопрос 4

Напишите программу, которая удаляет из строки все повторяющиеся символы.

#### Входные данные

На вход программы подаётся строка, содержащая символы таблицы ASCII.

#### Выходные данные

Программа должна вывести исходную строку, из которой удалены все повторяющиеся символы.

For example:

Input	Result
abc13a1b2z3	abc13a1b2z3
c	c
	abc132z

Answer:(penalty regime: 10, 20, ... %)

НЕВЕРНО:

Верно

<pre>print("".join(set(input())))</pre>	<pre>isstart = list(input()) isx = set(isstart) for i in isstart:     if i in isx:         isx.remove(i)     print(i, end= '')</pre>
<pre>a = input() b = set(a) #print(*b, sep='') # Неверно c='' for i in b:     c += i print(*c, sep='') #Неверно</pre>	
<pre>print(*set(list(input())), sep='')</pre>	

### Вопрос 5

Напишите программу, которая определяет количество различных символов, встречающихся в символьной строке.

#### Входные данные

На вход программы подаётся символьная строка.

Выходные данные

Программа должна вывести количество различных символов в этой строке.

**For example:**

Input	Result
aB122A	aB122A
B	B
	5

**Answer:(penalty regime: 10, 20, ... %)**

```
print(len(set(list(input()))))
```

## Программный тест. Списки и кортежи

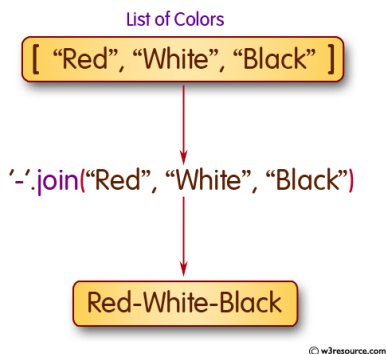
Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1. Напишите программу Python, которая принимает от пользователя последовательность чисел, разделенных запятыми, и генерирует список и кортеж с этими числами.

Решение

```
values = input("Input some comma seprated numbers : ")
list = values.split(",")
tuple = tuple(list)
print('List : ',list)
print('Tuple : ',tuple)
```

2. Напишите программу Python для объединения N строк.

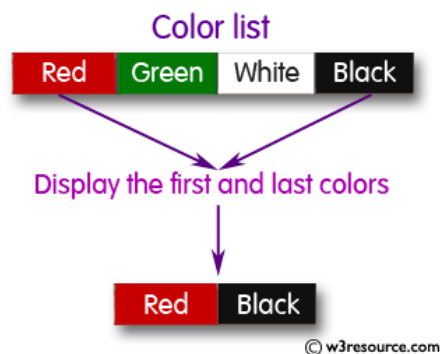


Решение

```
list_of_colors = ['Red', 'White', 'Black']
colors = '-'.join(list_of_colors)
print()
print("All Colors: "+colors)
print()
```

3. Напишите программу Python для отображения первого и последнего цветов из следующего списка.

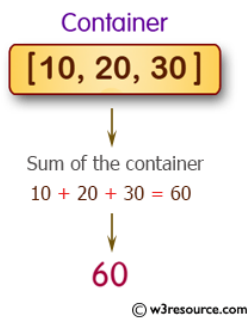
color\_list = ["Красный", "Зеленый", "Белый", "Черный"]



Решение

```
color_list = ["Red", "Green", "White", "Black"]
print( "%s %s"%(color_list[0],color_list[-1]))
```

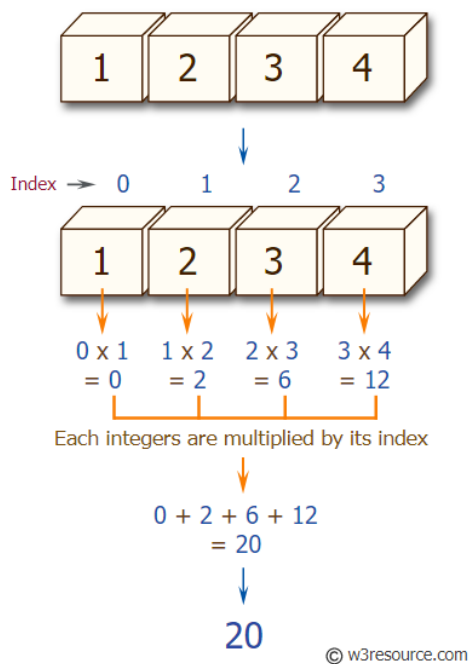
4. Напишите программу для вычисления суммы списка.



Решение

```
s = sum([10,20,30])
print("\nSum of the container: ", s)
print()
```

5. Напишите программу Python для вычисления суммы всех элементов данного массива целых чисел, где каждое целое число умножается на его индекс. Верните 0, если числа нет.



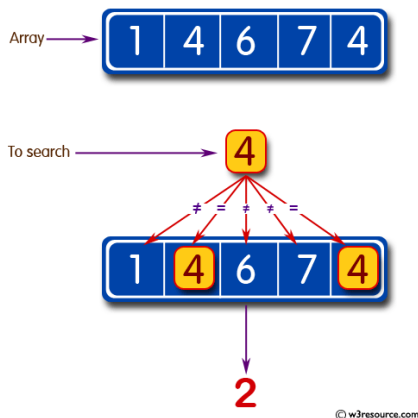
Решение

```
nums=list(map(int,input().split()))
print(sum(j*i for i, j in enumerate(nums)))
```

Другое решение

```
nums=list(map(int,input().split()))
s=0
for i in range(len(nums)):
    s+=i*nums[i]
print(s)
```

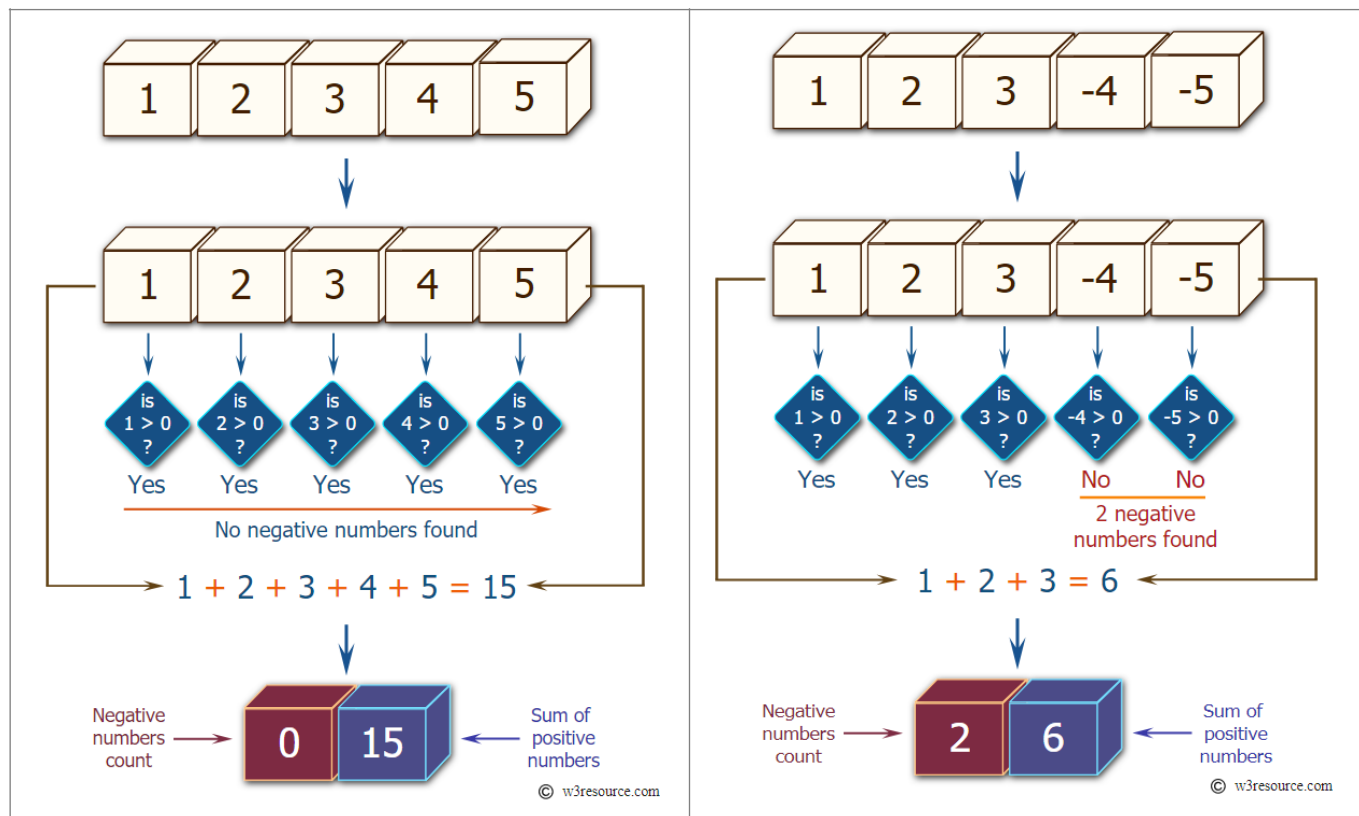
6. Напишите программу Python для подсчета числа 4 в данном списке



Решение

```
nums=list(map(int,input().split()))
count = 0
for num in nums:
    if num == 4:
        count = count + 1
print(count)
```

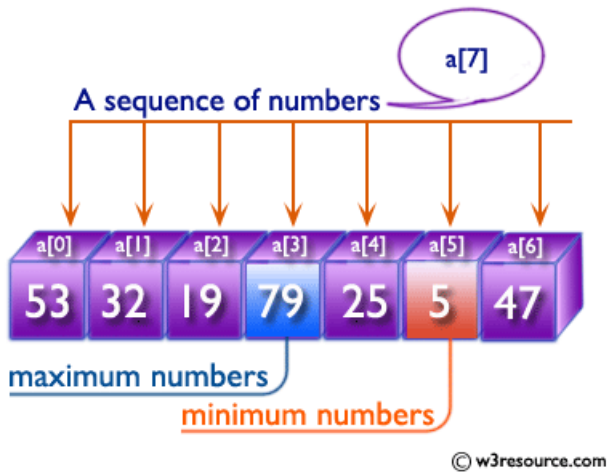
7. Напишите программу Python, которая принимает список чисел и создает список для хранения количества отрицательных чисел в первом элементе и суммы положительных чисел во втором элементе.



Решение

```
data=list(map(int,input().split()))
l = data[0]
s = data[0]
for num in data:
    if num> l:
        l = num
    elif num< s:
        s = num
print(l, s)
```

8. Напишите функцию Python, чтобы найти максимальное и минимальное числа из последовательности чисел. Примечание: не используйте встроенные функции.



Решение

```
nums=list(map(int,input().split()))
max_val = nums[1]
for i in range(len(nums)):
    for j in range(i+1, len(nums)):
        for k in range(j+1, len(nums)):
            max_val = max(nums[i] * nums[j] * nums[k], max_val)
print(max_val)
```

### Двумерные матрицы

1. Напишите программу Python для печати заданной матрицы чисел N на M

Решение

```
a = [[1, 2, 3], [4, 5, 6]]
for i in range(len(a)):
    print(a[i])
```

2. Найдите индексы первого вхождения максимального элемента.

#### Входные данные

Программа получает на вход размеры массива nn и mm, затем nn строк по mm целых чисел в каждой.

#### Выходные данные

Выведите два числа: номер строки и номер столбца, в которых стоит наибольший элемент в двумерном массиве. Если таких элементов несколько, то выводится тот, у которого меньше номер строки, а если номера строк равны то тот, у которого меньше номер столбца.

Примеры

#### ВХОДНЫЕ ДАННЫЕ

```
3 4
0 3 2 4
2 3 5 5
5 1 2 3
```

#### ВЫХОДНЫЕ ДАННЫЕ

```
1 2
```

Решение

```
n, m = map(int, input().split())
matrix = []
for i in range(n):
    matrix.append(list(map(int, input().split())))
maxi = matrix[0][0]
a = 0
b = 0
for i in range(n):
```

```

for j in range(m):
    if matrix[i][j] > maxi:
        a = i
        b = j
        maxi = matrix[i][j]
print(a, b)

```

3. Дано нечетное число  $n$ . Создайте двумерный массив из  $n \times n$  элементов, заполнив его символами "." (каждый элемент массива является строкой из одного символа). Затем заполните символами "\*" среднюю строку массива, средний столбец массива, главную диагональ и побочную диагональ. В результате единицы в массиве должны образовывать изображение звездочки. Выведите полученный массив на экран, разделяя элементы массива пробелами.

Примеры

ВХОДНЫЕ ДАННЫЕ

5

ВЫХОДНЫЕ ДАННЫЕ

```

* . * . *
. * * * .
* * * * *
. * * * .
* . * . *

```

Решение

```

n = int(input())
mas = [['.' * n for i in range(n)]
for i in range(n):
    for j in range(n):
        if i == (n // 2) or j == (n // 2):
            mas[i][j] = '*'
        elif i == j:
            mas[i][j] = '*'
        elif i + j == n - 1:
            mas[i][j] = '*'
for i in range(n):
    print(*mas[i])

```

4. Проверьте, является ли двумерный массив симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний.

Входные данные

Программа получает на вход число  $n$ , являющееся числом строк и столбцов в массиве. Далее во входном потоке идет  $n$  строк по  $n$  чисел, являющихся элементами массива.

Выходные данные

Программа должна выводить слово `yes` для симметричного массива и слово `no` для несимметричного.

Примеры

ВХОДНЫЕ ДАННЫЕ

3

0 1 2

1 5 3

2 3 4

ВЫХОДНЫЕ ДАННЫЕ

yes

#### ВХОДНЫЕ ДАННЫЕ

```
3
0 0 0
0 0 0
1 0 0
```

#### ВЫХОДНЫЕ ДАННЫЕ

no

Решение

```
n = int(input())
a = []
for i in range(n):
    a.append([int(j) for j in input().split()])
s = 0
for k in range(n):
    for m in range(n):
        if a[k][m] != a[m][k]:
            s = 1
            break
if s == 1:
    print('no')
else:
    print('yes')
```

5. Даны два числа  $n$  и  $m$ . Создайте двумерный массив  $[n][m]$  и заполните его по

следующим правилам: Числа, стоящие в строке 0 или в столбце 0 равны 1 ( $A[0][j]=1$ ,  $A[i]$

$[0]=1$ ). Для всех остальных элементов массива  $A[i][j]=A[i-1][j]+A[i][j-1]$ , то есть каждый элемент равен сумме двух элементов, стоящих слева и сверху от него.

Входные данные

Программа получает на вход два числа  $n$  и  $m$ .

Выходные данные

Выведите данный массив.

Примеры

#### ВХОДНЫЕ ДАННЫЕ

```
3 3
```

#### ВЫХОДНЫЕ ДАННЫЕ

```
1 1 1
1 2 3
1 3 6
```

Решение

```
n,m = map(int,input().split())
s = []
for i in range(n):
    s.append(m*[1])
for i in range(1,n):
    for j in range(1,m):
        s[i][j] = s[i-1][j]+s[i][j-1]
for i in range(len(s)):
    print(*s[i])
```





## Тест-задание 8. Списки и кортежи

### Вопрос 1

Выведите все элементы списка с четными индексами (то есть  $A[0]$ ,  $A[2]$ ,  $A[4]$ , ...).

Программа должна быть эффективной и не выполнять лишних действий!

Входные данные

Вводится список чисел. Все числа списка находятся на одной строке.

Выходные данные

Выведите ответ на задачу.

**For example:**

Input	Result
1 2 3 4 5	1 2 3 4 5  1 3 5

**Answer:(penalty regime: 10, 20, ... %) (Оба решения верны)**

<pre>a = input().split() for i in range(0, len(a), 2):     print(a[i], ' ', end='')</pre>	<pre>a = list(map(int, input().split())) for i in range(0, len(a), 2):     print(a[i], end=" ")</pre>
---	---

### Вопрос 2

Напишите программу, которая заполняет массив и определяет количество положительных трёхзначных чисел в этом массиве, которые не делятся на 5.

Входные данные

Входная строка содержит три числа: границы диапазона случайных чисел  $A$  и  $B$ , а также размер массива  $N$ . Все числа разделены пробелами. Гарантируется, что  $0 < N \leq 10000$ .

Выходные данные

В первой строке программа должна вывести  $N$  элементов построенного массива, разделив их пробелами, а во второй строке – количество трёхзначных чисел в массиве, которые не делятся на 5.

**For example:**

Input	Result
92 138 91 161 185 135 93 182 53 82	92 138 91 161 185 135 93 182 53 82  3

**Answer:(penalty regime: 10, 20, ... %)**

<pre>import random k = 0 a,b,n = map(int,input().split()) mas = [random.randint(a,b) for i in range (n)] for i in range (n):     if ((mas[i] &gt;=100) and (mas[i] &lt; 1000) and (mas[i]%5 &gt; 0)):         k+=1 print(*mas) print(k)</pre>	верно (у них не подключен модуль random)
---	--

## Вопрос 2 - НОВЫЙ

Напишите программу, которая заполняет массив и определяет количество положительных трёхзначных чисел в этом массиве, которые не делятся на 5.

### Входные данные

Входная строка содержит массив .

### Выходные данные

В первой строке программа должна вывести – количество трёхзначных чисел в массиве, которые не делятся на 5.

**For example:**

Input	Result
92 138 91 161 185 135 93 182 53 82	92 138 91 161 185 135 93 182 53 82
	3

**Answer:(penalty regime: 10, 20, ... %)**

<pre>mas = [int(i) for i in input().split()] k=0 for i in range (len(mas)):     if ((mas[i] &gt;=100) and (mas[i] &lt; 1000) and (mas[i]%5 &gt; 0)):         k+=1 #print(*mas) print(k)</pre>	<pre>a = list(map(int, input().split())) s=0 for i in range(len(a)):     if a[i]&gt;99 and a[i]&lt;1000 and a[i]%5!=0:         s+=1 print(s)</pre>
---	--

## Вопрос 3

В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

### Входные данные

Вводится список целых чисел. Все числа списка находятся на одной строке.

### Выходные данные

Выведите ответ на задачу.

**Answer:(penalty regime: 10, 20, ... %)**

```

a = [int(s) for s in input().split()]
index_of_min = 0
index_of_max = 0
for i in range(1, len(a)):
    if a[i] > a[index_of_max]:
        index_of_max = i
    if a[i] < a[index_of_min]:
        index_of_min = i
a[index_of_min], a[index_of_max] =
a[index_of_max], a[index_of_min]
print(' '.join([str(i) for i in a]))

```

```

abc = list(map(int, input().split()))
x = abc.index(max(abc))
y = abc.index(min(abc))
abc[x], abc[y] = abc[y], abc[x]
print(*abc)

```

#### Вопрос 4

Напишите программу, которая считывает массив определяет максимальный среди чётных положительных элементов этого массива.

##### Входные данные

Входная строка содержит три числа: границы диапазона случайных чисел  $A$  и  $B$ , а также размер массива  $N$ . Все числа разделены пробелами. Гарантируется, что  $0 < N \leq 10000$ .

##### Выходные данные

В первой строке программа должна вывести  $N$  элементов построенного массива, разделив их пробелами, а во второй строке – максимальный из чётных положительных элементов этого массива. Если в массиве нет таких элементов, программа должна вывести число -1.

**Answer:(penalty regime: 10, 20, ... %)**

```

import random
k = 0
a,b,n = map(int,input().split())
mas = [random.randint(a,b) for i in
range (n)]
res = 0
for i in range (n):
    if (mas[i] > res):
        res = mas[i]
print(' '.join([str(i) for i in mas]))
if res > 0:
    print(res)
else:
    res=-1
    print(res)

```

верно (у них не подключен модуль random)

#### Вопрос 4 - НОВЫЙ

Напишите программу, которая считывает массив определяет максимальный среди чётных положительных элементов этого массива.

##### Входные данные

Входная строка массив. Все числа разделены пробелами. Гарантируется, что  $0 < N \leq 10000$ .

##### Выходные данные

В первой строке программа должна вывести максимальный из чётных положительных элементов этого массива. Если в массиве нет таких элементов, программа должна вывести число -1.

**For example:**

Input	Result
0 3 86 20 27 67 32 16 37 42	0 3 86 20 27 67 32 16 37 42 86

**Answer:(penalty regime: 10, 20, ... %)**

<pre>mas = [int(i) for i in input().split()] res = 0 for i in range (len(mas)):     if (mas[i] &gt; res):         res = mas[i] if res &gt; 0:     print(res) else:     res=-1     print(res)</pre>	<pre>A = list(map(int, input().split())) M = - 999999999999 for i in range(len(A)):     if A[i] &gt; M and A[i]% 2 == 0 and A[i] &gt; 0:         M = A[i] if M == - 999999999999-1:     M = -1 print(M)</pre>
--	---

## Вопрос 5

Дан список. Не изменяя его и не используя дополнительные списки, определите, какое число в этом списке встречается чаще всего.

Если таких чисел несколько, выведите любое из них.

Входные данные

Вводится список чисел. Все числа списка находятся на одной строке.

Выходные данные

Выведите ответ на задачу.

**For example:**

Input	Result
1 2 3 2 3 3	1 2 3 2 3 3 3
1 2 3 4 5 6 7 8 9 1	1 2 3 4 5 6 7 8 9 1 1

**Answer:(penalty regime: 10, 20, ... %)**

<pre>a=list(map(int,input().split())) print(max(a, key=a.count))</pre>	<pre>a = [int(i) for i in input().split()] b = 0 c = 0 for i in range(0, len(a)):     if a.count(a[i]) &gt; c:         c = a.count(a[i])         b = a[i] print(b)</pre>
--	--

## Вопрос 6

Переставьте соседние элементы списка ( $A[0]$  с  $A[1]$ ,  $A[2]$  с  $A[3]$  и т.д.). Если элементов нечетное число, то последний элемент остается на своем месте.

### Входные данные

Вводится список чисел. Все числа списка находятся на одной строке.

### Выходные данные

Выведите ответ на задачу.

### For example:

Input	Result
1 2 3 4 5	1 2 3 4 5 2 1 4 3 5

**Answer:(penalty regime: 10, 20, ... %)**

```
a=list(map(int,input().split()))
for i in range(1, len(a), 2):
    a[i - 1], a[i] = a[i], a[i - 1]
print(' '.join([str(i) for i in a]))
```

## Программный тест 12. Словари

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1) Напишите программу Python для перебора словарей с использованием циклов for.

```
d = {'x': 10, 'y': 20, 'z': 30}
for dict_key, dict_value in d.items():
    print(dict_key, '->', dict_value)
```

2) Напишите программу Python создания и печати словаря, содержащего число (от 1 до n) в форме (x, x \* x).

```
n=int(input("Input a number "))
d = dict()
for x in range(1,n+1):
    d[x]=x*x print(d)
```

3) Напишите программу Python для суммирования всех элементов словаря.

```
my_dict = {'data1':100,'data2':-54,'data3':247}
print(sum(my_dict.values()))
```

4) Напишите программу Python для удаления ключа из словаря.

```
myDict = {'a':1, 'b':2, 'c':3, 'd':4}
print(myDict)
if 'a' in myDict:
    del myDict['a']
print(myDict)
```

5) В единственной строке записан текст. Для каждого слова из данного текста подсчитайте, сколько раз оно встречалось в этом тексте ранее.

Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.

Входные данные

one two one tho three

Выходные данные

0 0 1 0 0

# В словарях существует метод .get(). В обычном случае, если вы вызываете несуществующий ключ name\_dict['key'], вы получите исключение KeyError.

Однако, если вызвать ключ через метод d.get('key'), то исключения не будет и, если ключа нет, то словарь возвратит None.

Если вы хотите назначить переменную вместо отсутствующего ключа, то можно назначить второй параметр: d.get('key', 0).

Лучше всего это применять при переборе числовых ключей:  $A[w] = A.get(w, 0) + 1$

```
t=input().split() #считаем строку
for w in t: # пройдем по элементам слов строки
    A[w] = A.get(w, 0) + 1 # Изменим значение словаря на единицу, если по ключу не найдено значение
присвоим значение 0
print(A[w] - 1, end=' ')
```

6) Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны.

Для слова из словаря, записанного в последней строке, определите его синоним.

```
n = int(input())
p = {}
p = dict(input().split() for j in range(n))
k = input()
for key, value in p.items():
    print(key, '->', value)
```

7) Напишите программу Python для сортировки словаря по ключу.

```
color_dict = {'red': '#FF0000', 'green': '#008000', 'black': '#000000',
              'white': '#FFFFFF'}
for key in sorted(color_dict):
    print("%s: %s" % (key, color_dict[key]))
```



## Тест-задание 12. Словари

### Вопрос 1

В единственной строке записан текст. Для каждого слова из данного текста подсчитайте, сколько раз оно встречалось в этом тексте ранее.

Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.

For example:

Input	Result
one two one tho three	one two one tho three  0 0 1 0 0

Answer:(penalty regime: 10, 20, ... %)

```
t=input().split() #считаем строку
A={}
for w in t: # пройдем по элементам слов строки
    A[w] = A.get(w, 0) + 1 # Изменим значение словаря на единицу, если по ключу
    не найдено значение присвоим значение 0
    print(A[w] - 1, end=' ')
```

### Вопрос 2

Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны.

Для слова из словаря, записанного в последней строке, определите его синоним.

Answer:(penalty regime: 10, 20, ... %)

```
n = int(input())
d = {}
for i in range(n):
    first, second = input().split()
    d[first] = second
    d[second] = first
print(d[input()])
```

### Вопрос 3

Как известно, в США президент выбирается не прямым голосованием, а путем двухуровневого голосования. Сначала проводятся выборы в каждом штате и определяется победитель выборов в данном штате. Затем проводятся государственные выборы: на этих выборах каждый штат имеет определенное число голосов — число выборщиков от этого штата. На практике, все выборщики от штата голосуют в соответствии с результатами голосования внутри штата, то есть на заключительной стадии выборов в голосовании участвуют штаты, имеющие различное число голосов.

В первой строке дано количество записей. Далее, каждая запись содержит фамилию кандидата и число голосов, отданных за него в одном из штатов. Подведите итоги выборов: для каждого из участника голосования определите число отданных за него голосов. Участников нужно выводить в алфавитном порядке.

Answer:(penalty regime: 10, 20, ... %)

```
num_votes = {}
```

```
for _ in range(int(input())):
    candidate, votes = input().split()
    num_votes[candidate] = num_votes.get(candidate, 0) + int(votes)

for candidate, votes in sorted(num_votes.items()):
    print(candidate, votes)
```

## Программный тест 10. Методы списков и строк

### Методы типа строки

С помощью метода <b>len()</b> можно узнать количество символов в строке	<pre>&gt;&gt;&gt; len('Python') 6  string1 = "PROGRAMIZ" print(string1[len(string1) - 2]) #l print(string1[5-len(string1)]) #A</pre>
<p><b>str.count(sub, start = 0,end = len(string))</b> - возвращает количество вхождений подстроки sub в диапазоне [start, end]. Необязательные аргументы start и end интерпретируются как срез.</p> <ul style="list-style-type: none"><li>• <b>sub</b> – это подстрока для поиска.</li><li>• <b>start</b> – поиск начинается с этого индекса. Первый символ начинается с 0 индекса. По умолчанию поиск начинается с 0 индекса.</li><li>• <b>end</b> – поиск заканчивается на этом индексе. Первый символ начинается с 0 индекса. По умолчанию поиск заканчивается на последнем индексе.</li></ul> <p>Возвращает количество вхождений.</p>	
<b>find(str, [start],[end])</b> - Поиск подстроки в строке. Возвращает номер первого вхождения или -1	<pre>&gt;&gt;&gt; s = 'PythonohtyP' &gt;&gt;&gt; s.find('t') 2  #вернуть позицию второго вхождения данной строки txt="the quick brown fox jumps over the lazy dog" str1="the" print(txt.find(str1, txt.find(str1)+1)) Результат: 31</pre>
<b>rfind(str, [start],[end])</b> - Поиск подстроки в строке. Возвращает номер последнего вхождения или -1	<pre>&gt;&gt;&gt; s = 'PythonohtyP' &gt;&gt;&gt; s.rfind('t') 8</pre>
<b>index(str, [start],[end])</b> - Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает <b>ValueError</b>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.index('t') 2</pre>
<b>rindex(str, [start],[end])</b> - Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает <b>ValueError</b>	<pre>&gt;&gt;&gt; s = 'PythonohtyP' &gt;&gt;&gt; s.rindex('t') 8</pre>
<b>replace(шаблон, замена)</b> - Замена шаблона	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.replace('P', 'AAA') 'AAAYthon'</pre>
<b>split(символ)</b> - Разбиение строки по разделителю  Поведение метода аналогично поведению <b>str.rsplit</b> за исключением направления разбиения строки.	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.split('t') ['Py', 'hon']</pre>

<p><b>str.rsplit(sep = None, maxsplit = -1)</b> – Функция разбивает строку на части, возвращает список слов заданной строки с использованием разделителя в качестве строки-разделителя. Направление разбиения: справа налево.</p> <ul style="list-style-type: none"> <li>• Если задано <i>maxsplit</i>, список будет содержать не более <i>maxsplit</i> + 1 элементов.</li> <li>• Если <i>maxsplit</i> не указан или -1, то количество разделений не ограничено.</li> <li>• Если задано <i>sep</i>, последовательные разделители не группируются вместе и считаются ограничивающими пустые</li> </ul> <p>Поведение метода аналогично поведению <i>str.split</i> за исключением направления разбиения строки.</p>	<pre>filename = input("Input the Filename: ") f_extns = filename.split(".")  print ("The extension of the file is : " + repr(f_extns[-1]))</pre>
<p><b>join(список)</b> - Сборка строки из списка с разделителем <b>S</b></p> <p>(Объединяет элементы списка, добавляя тот символ, который указан перед join)</p>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.join(['a','b','c']) 'aPythonbPythonc'</pre> <pre>a=['2', '12', '85', '06'] print('*.join(a)    # 2*12*85*06</pre>
<p><b>str.ljust(width[, fillchar])</b> - возвращает строку с равнением по левому краю. Заполнение выполняется с использованием указанного FillChar (по умолчанию пробел). Исходная строка возвращается, если ширина меньше, чем указанная длина строки.</p> <ul style="list-style-type: none"> <li>• width – это длина строки в общей сложности после заполнения.</li> <li>• fillchar – является заполнителем, по умолчанию пробел.</li> </ul>	<pre>str = "это пример строки....wow!!!" print str.ljust(45, '*')</pre> <p>Результат</p> <pre>это пример строки....wow!!!*****</pre>
<p><b>str.rjust(width[, fillchar])</b> – аналогична <i>ljust()</i>, но с равнением по правому краю</p>	<pre>str = "это строковый пример....wow!!!" print (str.rjust(50, '*'))</pre> <p>Результат</p> <pre>*****это строковый пример....wow!!!</pre>
<p><b>isdigit()</b> - Состоит ли строка из цифр</p>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.isdigit() False</pre>
<p><b>isalpha()</b> - Состоит ли строка из букв</p>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.isalpha() True</pre>
<p><b>isalnum()</b> - Состоит ли строка из цифр или букв</p>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.isalnum() True</pre>
<p><b>islower()</b> - Состоит ли строка из символов в нижнем регистре</p>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.islower() False</pre> <p>#существуют ли прописные буквы в строке?</p> <pre>str1 = 'A8238i823acdeOUEI' print(any(c.islower() for c in str1))</pre>

<b>isupper()</b> - Состоит ли строка из символов в верхнем регистре	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.isupper() False</pre>
<b>istitle()</b> - Начинаются ли слова в строке с заглавной буквы	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.istitle() True</pre>
<b>upper()</b> - Преобразование строки к верхнему регистру	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.upper() 'PYTHON'</pre>
<b>lower()</b> - Преобразование строки к нижнему регистру	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.lower() 'python'</pre>
<b>capitalize()</b> - Делает первое слово предложения с большой буквы	<pre>s = 'python' print(s.capitalize()) # Python</pre>
<b>startswith(str)</b> - Начинается ли строка <b>S</b> с шаблона <b>str</b>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.startswith('P') True</pre>
<b>endswith(str)</b> - Заканчивается ли строка <b>S</b> шаблоном <b>str</b>	<pre>&gt;&gt;&gt; s = 'Python' &gt;&gt;&gt; s.endswith('a') False</pre>

Пример использования методов строк:

```
>>> "PrOgRaMiZ".lower()
'programiz'
>>> "PrOgRaMiZ".upper()
'PROGRAMIZ'
>>> "This will split all words into a list".split()
['This', 'will', 'split', 'all', 'words', 'into', 'a', 'list']
>>> ' '.join(['This', 'will', 'join', 'all', 'words', 'into', 'a', 'string'])
'This will join all words into a string'
>>> 'Happy New Year'.find('ew')
7
>>> 'Happy New Year'.replace('Happy', 'Brilliant')
'Brilliant New Year'
```

## Методы списка Python

Методы, которые доступны со списком объектов в программировании на Python, приведены в таблице ниже. Они доступны как `list.method()`. Некоторые из методов уже были использованы выше.

Методы списка Python

**append ()** - добавляет элемент в конец списка

**extend ()** - добавляет все элементы списка в другой список

**insert ()** - вставить элемент по указанному индексу

**remove ()** - удаляет элемент из списка

**pop ()** - удаляет и возвращает элемент по указанному индексу

**clear ()** - удаляет все элементы из списка

**index ()** - возвращает индекс первого соответствующего элемента

**count ()** - Возвращает количество элементов, переданных в качестве аргумента.

**sort ()** - сортировка элементов в списке в порядке возрастания

**reverse ()** - обратный порядок элементов в списке

`copy()` - возвращает поверхностную копию списка.  
Некоторые примеры методов списка Python:

```
my_list = [3, 8, 1, 6, 0, 8, 4]

# Выведет: 1
print(my_list.index(8))

# Выведет: 2
print(my_list.count(8))

my_list.sort()

# Результат отсортированный список: [0, 1, 3, 4, 6, 8, 8]
print(my_list)

my_list.reverse()

# Результат переворота: [8, 8, 6, 4, 3, 1, 0]
print(my_list)
```

Результат:

```
1
2
[0, 1, 3, 4, 6, 8, 8]
[8, 8, 6, 4, 3, 1, 0]
```

Количество элементов в списке можно узнать с помощью функции `len()` :

```
>>> a = ['P', 'y', 't', 'h', 'o', 'n']
>>> len(a)
6
```

Метод `copy()` - поверхностная копия списка, при присвоении переменной копии списка, значение данного списка не изменяется в случае изменения первого. Если переменной присвоить список через " =", тогда значение этой переменной будет меняться при изменении оригинала.

```

>>> var = ['l', 'i', 's', 't']
>>> asd = var.copy()
>>> print(asd)
['l', 'i', 's', 't']
>>> var = ['l', 'i', 's', 't']
>>> asd = var
>>> print(asd)
['l', 'i', 's', 't']
>>> print(var)
['l', 'i', 's', 't']
>>> var.reverse()
>>> print(asd)
['t', 's', 'i', 'l']
>>> print(var)
['t', 's', 'i', 'l']
>>> var = ['l', 'i', 's', 't']
>>> asd = var.copy()
>>> print(asd)
['l', 'i', 's', 't']
>>> print(var)
['l', 'i', 's', 't']
>>> var.reverse()
>>> print(asd)
['l', 'i', 's', 't']
>>> print(var)
['t', 's', 'i', 'l']

```

Мы можем проверить, присутствие элементов в списке, используя ключевое слово in:

```
my_list = ['p', 'r', 'o', 'b', 'l', 'e', 'm']
```

```

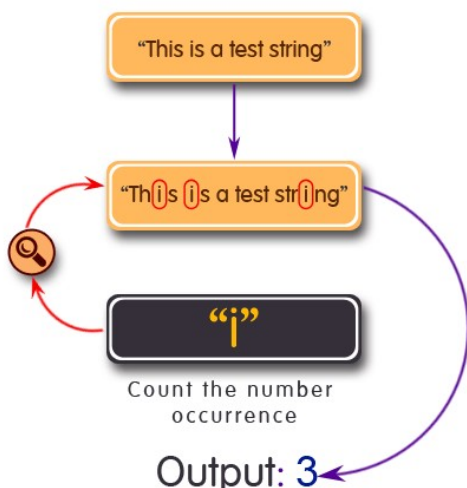
# Выведет: True
print('p' in my_list)

# Выведет: False
print('a' in my_list)

# Выведет: True
print('c' not in my_list)

```

1. Напишите программу Python для подсчета количества вхождений определенного символа в строку.



Решение

```
s = "The quick brown fox jumps over the lazy dog."  
print()  
print(s.count("q"))  
print()
```

2. Решите с помощью генераторов списков. Дан список, заполненный произвольными целыми числами. Получить новый список, элементы которого будут квадратами элементов исходного списка [1, 2, 4, 0] --> [1, 4, 16, 0]

Решение

```
list1 = [-1, 2, 3, 4, 5, 6, 7, -4]  
list2 = [i ** 2 for i in list1]  
print(list2)
```

3. Решите с помощью генераторов списков!

Дан список, заполненный произвольными числами. Получить список из элементов исходного, удовлетворяющих следующим условиям:

- Элемент кратен 3
- Элемент положительный
- Элемент не кратен 4

Решение

```
import random  
l = [random.randint(-5, 20) for _ in range(10)]  
print(l)  
result = [x for x in l if x % 3 == 0 and x >= 0 and x % 4 != 0]  
print(result)
```



## Тест-задание 10. Методы списков и строк

### Вопрос 1

Напишите программу, которая заменяет в символьной строке все буквы а на буквы b (латинские, строчные).

#### Входные данные

Входная строка содержит символы латинского алфавита, как строчные, так и заглавные.

#### Выходные данные

Программа должна заменить во всей строке строчные буквы а на буквы b и вывести в первой строке получившуюся символьную строку, а во второй – количество выполненных замен.

#### For example:

Input	Result
aabbccabade f	aabbccabade f  bbbbccbbdde f  4

Answer:(penalty regime: 10, 20, ... %)  
работало

Не уверен, раньше

<pre>a = list(input()) k=0 n=len(a) for i in range(n):     if a[i] == 'a':         a.pop(i)         a.insert(i, 'b')         k+=1 print(*a, sep='') print(k)</pre>	<pre>s = input() sa = s.count('a') s1 = s.replace('a', 'b') print(s1) print(sa)</pre>
--	---

### Вопрос 2

Дана строка, состоящая из слов, разделенных пробелами. Определите, сколько в ней слов. Используйте для решения задачи метод `count`.

#### Входные данные

Вводится строка.

#### Выходные данные

Выведите ответ на задачу.

#### For example:

Input	Result
Hello world	Hello world  2

Answer:(penalty regime: 10, 20, ... %)

```
a = list(input())
```

```
k=a.count(' ')+1
print(k)
```

### Вопрос 3

Дана строка. Удалите из этой строки все символы @.

**For example:**

Input	Result
Bilbo.Baggins@bagend.hobbiton.shire.me	Bilbo.Baggins@bagend.hobbiton.shire.me Bilbo.Bagginsbagend.hobbiton.shire.me

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

<pre>a = list(input()) while '@' in a:     a.remove('@') print(*a, sep='')</pre>	<pre>x=input() x=x.replace("@","") print(x)</pre>
--	---

### Вопрос 4

Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.

Входные данные

Вводится список чисел. Все числа списка находятся на одной строке.

Выходные данные

Выведите ответ на задачу.

**For example:**

Input	Result
1 2 3 2 1	1 2 3 2 1 3 2

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

<pre>a = [int(i) for i in input().split()] ind=0 outp=a[0] for i in range(len(a)):     if a[i] &gt; outp:         outp = a[i]         ind = i print(outp, ind, sep='')</pre>	<pre>A = list(map(int, input().split())) print(max(A), A.index(max(A)))</pre>
--	---

### Вопрос 5

На вход программе подается одна строка – числа, разделенные пробелами. Считайте эти числа в список. Отсортируйте элементы списка по возрастанию и по убыванию. Выведите полученные списки на экран.

**Answer:(penalty regime: 10, 20, ... %)**

```
a=[int(i) for i in input().split()]
print(*sorted(a))
print(*sorted(a, reverse = True))
```

#### Вопрос 6

Используя генератор списков.

Напишите программу, которая заполняет массив первыми  $N$  натуральными числами и выводит его.

Входные данные

Входная строка содержит размер массива  $N$ . Гарантируется, что  $0 < N \leq 10000$ .

Выходные данные

Программа должна вывести содержимое массива:  $N$  последовательных натуральных чисел от 1 до  $N$ .

**For example:**

Input	Result
5	5 1 2 3 4 5

**Answer:(penalty regime: 10, 20, ... %) (оба верные)**

```
n=int(input())
arr = [i for i in range(1, n+1)]
print(*arr)
```

```
array = [n for n in range(1,
int(input())+1)]
print(*array)
```

#### Вопрос 7

Используя генератор списков.

Напишите программу, которая заполняет массив первыми  $N$  натуральными числами в обратном порядке (начиная с последнего) и выводит его.

Входные данные

Входная строка содержит размер массива  $N$ . Гарантируется, что  $0 < N \leq 10000$ .

Выходные данные

Программа должна вывести содержимое массива:  $N$  последовательных натуральных чисел от  $N$  до 1 в обратном порядке.

**For example:**

Input	Result
5	5 5 4 3 2 1

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

```
n=int(input())
arr = [i for i in range(n, 0, -1)]
print(*arr)
```

```
n = int(input())
lst = reversed(range(1, 1 + n))
print(*lst)
```

### Вопрос 8

Используя генератор списков.

Напишите программу, которая заполняет массив из  $N$  элементов в обратном порядке последовательными натуральными числами, начиная с последнего элемента, который должен быть равен  $X$ .

#### Входные данные

Входная строка содержит два целых числа: начальное значение  $X$  и размер массива  $N$ , разделённые пробелом. Гарантируется, что  $0 < N \leq 10000$ .

#### Выходные данные

Программа должна вывести содержимое массива: убывающую последовательность из  $N$  последовательных натуральных чисел; последний элемент должен быть равен  $X$ .

#### For example:

Input	Result
6 4	6 4
	9 8 7
	6

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

```
a = [int(i) for i in input().split()]
arr = [i for i in range(a[0]+a[1]-1, a[0]-1, -1)]
print(*arr)
```

```
n, m = map(int, input().split())
lst = reversed(range(n, m + n))
print(*lst)
```

### Вопрос 9

Используя генератор списков.

Напишите программу, которая заполняет массив из  $N$  элементов последовательными натуральными числами, начиная с  $X$ , и выводит его.

#### Входные данные

Входная строка содержит два целых числа: начальное значение  $X$  и размер массива  $N$ , разделённые пробелом. Гарантируется, что  $0 < N \leq 10000$ .

#### Выходные данные

Программа должна вывести содержимое массива:  $N$  последовательных натуральных чисел, начиная с  $X$ .

#### For example:

Input	Result
4 5	4 5
	4 5 6 7
	8

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

```
a = [int(i) for i in input().split()]
arr = [i for i in range(a[0], a[0]+a[1])]
print(*arr)
```

```
n, m = map(int, input().split())
lst = range(n, m + n)
print(*lst)
```

#### Вопрос 10

Используя генератор списков.

Напишите программу, которая заполняет массив квадратами натуральных чисел от 1 до  $N$ .

Входные данные

Входная строка содержит размер массива  $N$ . Гарантируется, что  $0 < N \leq 10000$ .

Выходные данные

Программа должна вывести содержимое массива: квадраты  $N$  первых натуральных чисел.

**For example:**

Input	Result
5	5 1 4 9 16 25

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

```
n=int(input())
arr = [i*i for i in range(1, n+1)]
print(*arr)
```

```
n = int(input())
print(*[i * i for i in range(1, n + 1)])
```

#### Вопрос 11

Используя генератор списков. Напишите программу, которая заполняет массив из  $N$  элементов степенями числа 2, начиная с  $2^1$  до  $2^N$ , в обратном порядке.

Входные данные

Входная строка содержит размер массива  $N$ . Гарантируется, что  $0 < N \leq 30$ .

Выходные данные

Программа должна вывести содержимое массива:  $N$  первых степеней числа 2 в обратном порядке (последний элемент должен быть равен  $2^1$ ).

**For example:**

Input	Result
5	5 32 16 8 4 2

**Answer:(penalty regime: 10, 20, ... %) (Оба верные)**

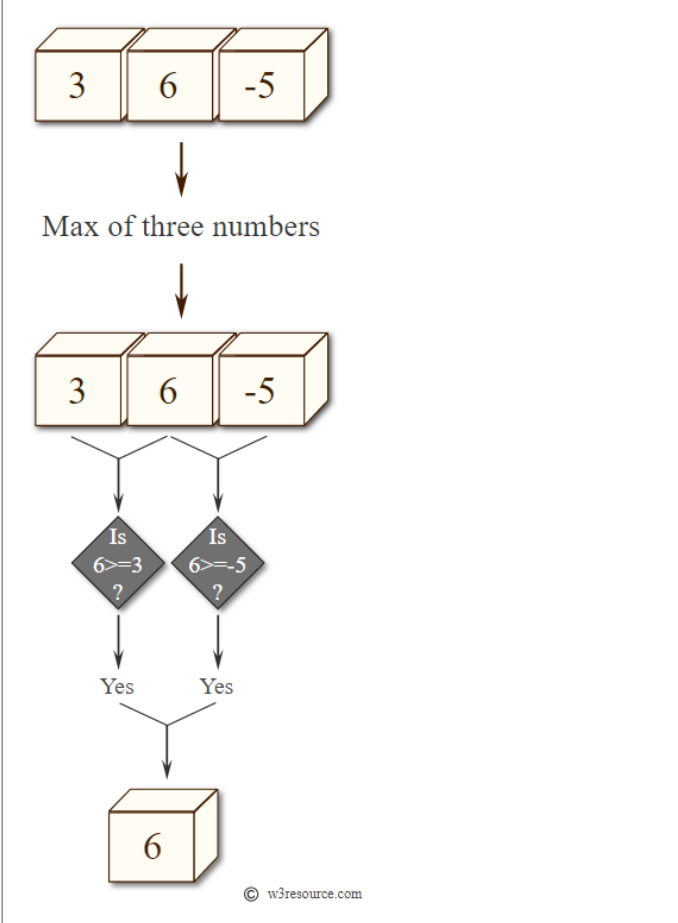
```
n=int(input())
arr = [2 ** i for i in range(n, 0,-1)]
print(*arr)
```

```
n=int(input())
a=[]
b=2
for i in range(n):
    a.append(b**(n-i))
print(*a)
```

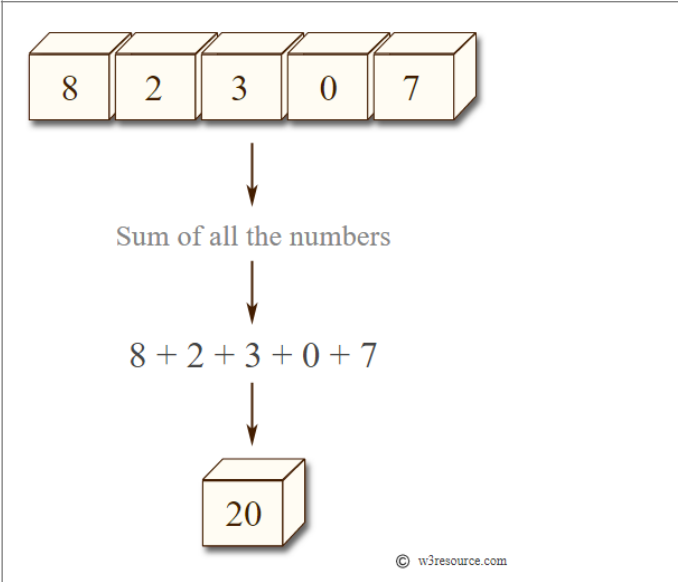
## Программный тест 13. Функции

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1. Напишите функцию Python, чтобы найти максимум трех чисел.

	<p>Решение</p> <pre>def max_of_two( x, y ):     if x &gt; y:         return x     else:         return y  def max_of_three( x, y, z ):     return max_of_two( x, max_of_two( y, z ) )  print(max_of_three(3, 6, -5))</pre>
--	--

2. Напишите функцию Python для суммирования всех чисел в списке.

	<p>Решение</p> <pre>def sum(numbers):     total = 0     for x in numbers:         total += x     return total  print(sum((8, 2, 3, 0, 7)))</pre>
---	--

## Тест-задание 13. Функции

### Вопрос 1

Напишите функцию, вычисляющую значение  $a^n$ .

Входные данные

Вводится 2 числа -  $a$  (вещественное) и  $n$  (целое неотрицательное).

Выходные данные

Необходимо вывести значение  $a^n$ .

**For example:**

Input	Result
2 1	2 1 2.0

**Answer:(penalty regime: 10, 20, ... %)**

```
def expon(a, n):  
    return (a**n)  
  
x, y = map(float, input().split())  
y=int(y)  
print(expon(x, y))
```

### Вопрос 2

Напишите функцию, находящую наименьшее из четырех данных чисел.

Входные данные

Вводится четыре числа.

Выходные данные

Необходимо вывести наименьшее из 4-х данных чисел.

**For example:**

Input	Result
4 5 6 7	4 5 6 7 4

**Answer:(penalty regime: 10, 20, ... %)**



```
def expon(a1, a2, a3, a4):
    return (min(a1, a2, a3, a4))
x, y, z, w = map(int, input().split())
print(expon(x, y, z, w))
```

```
def min2(a1, b1):
    if a1 <= b1:
        return a1
    else:
        return b1

def min4(a_, b_, c_, d_):
    return(min2(min2(a_, b_), min2(c_, d_)))

a, b, c, d = map(int,input().split())
print(min4(a, b, c, d))
```

### Вопрос 3

Напишите функцию

`def xor(x, y):`(Python) реализующую функцию "Исключающее ИЛИ" двух логических переменных `x` и `y`. Функция `Xor` должна возвращать `true`, если ровно один из ее аргументов `x` или `y`, но не оба одновременно равны `true`.

#### Входные данные

Вводится 2 числа - `x` и `y` (`x` и `y` равны 0 или 1, 0 соотвечает значению `false`, 1 соответствует значению `true`).

#### Выходные данные

Необходимо вывести 0 или 1 - значение функции от `x` и `y`.

#### For example:

Input	Result
0 0	0 0
	0

#### Answer:(penalty regime: 10, 20, ... %)

```
def xor(x, y):
    if (x != 0 and y!= 0) or (x == 0 and y == 0):
        return (0)
    else:
        return (1)
x, y = map(int, input().split())
print(xor(x, y))
```

```
def zena(s,f):
    if s==0 and f==0 or s==1 and f==1:
        print(0)
    else:
        print(1)

a,n=map(int,input().split())
(zena(a,n))
```

### Вопрос 4

Даны четыре действительных числа:  $x_1, y_1, x_2, y_2$ . Напишите функцию `distance(x1, y1, x2, y2)`, вычисляющую расстояние между точкой  $(x_1, y_1)$  и  $(x_2, y_2)$ . Считайте четыре действительных числа и выведите результат работы этой функции.

Входные данные

Вводятся четыре действительных числа.

Выходные данные

Выведите `ответ` на задачу.

**For example:**

Input	Result
0	0
0	0
1	1
1	1
	1.41421356237309
	51
0	0
0	0
1	1
0	0
	1.0

**Answer:(penalty regime: 10, 20, ... %)**

```
import math

def distance(x1, y1, x2, y2):
    return (math.sqrt((x2-x1)**2 + (y2-y1)**2))

a = float(input())
b = float(input())
c = float(input())
d = float(input())
print(distance(a, b, c, d))
```

**Вопрос 5**

Напишите программу, которая вычисляет количество цифр числа. Используйте функцию.

Входные данные

Входная строка содержит одно целое неотрицательное число.

Выходные данные

Программа должна вывести количество цифр полученного числа.

**For example:**

Input	Result
1234	1234
5	5
	5

**Answer:(penalty regime: 10, 20, ... %)**

```
def num_digits(n):
    return len(str(abs(n)))
print(num_digits(int(input())))
```

```
a = int(input())
count = 0
if a == 0:
    print(1)
else:
    while a > 0:
        a = a//10
        count+=1
    print(count)
```

## Вопрос 6

Напишите программу, которая "разворачивает" число наоборот, например, из числа 123 строит число 321. Используйте функцию для выполнения этой операции.

### Входные данные

Входная строка содержит одно натуральное число. Гарантируется, что число не оканчивается на 0.

### Выходные данные

Программа должна вывести число, в котором цифры стоят в обратном порядке в сравнении с исходным.

### For example:

Input	Result
123	123 321

Answer:(penalty regime: 10, 20, ... %)

```
def reverse_number(n):
    n=str(n)[::-1]
    n=int(n)
    return n
print(reverse_number(int(input())))
```

```
a=int(input())
def rev(a):
    i=0
    r=0
    b=a
    while b>0:
        b=b//10
        i+=1
    i-=1
    while a>0:
        b=a%10
        a=a//10
        r=r+b*10**i
        i-=1
    return(r)
print(rev(a))
```



## Программный тест 14. Лямбда-функции

В Python анонимная функция - это **функция**, которая определяется без имени.

В то время как обычные функции определяются с помощью **def** ключевого слова в Python, анонимные функции определяются с помощью **lambda** ключевого слова.

Следовательно, анонимные функции также называются лямбда-функциями.

Пример лямбда-функции в Python

Вот пример лямбда-функции, которая удваивает входное значение.

```
# Программа показывает использование lambda функции
double = lambda x: x * 2

print(double(5))
```

Вывод: 10

По сути это краткая запись функции:

```
def double (x): return x * 2
```

Итак:

Функция в Python может быть определена через классический оператор **def**:

```
def add_one (a, b):
    return a + b + 1
```

А можно эту запись заменить эквивалентным лямбда-выражением:

```
add_one = lambda a, b: a + b + 1
```

Это второй способ определения функций.

В Python мы обычно используем его в качестве аргумента функции высшего порядка (функции, которая принимает другие функции в качестве **аргументов**). Лямбда-функции используются вместе со встроенными функциями, такими как **filter()** и **map()** т. д.

**map()** - принимает функцию-аргумент и применяет её ко всем элементам входящей последовательности.

```
# напечатаем квадраты чисел от 1 до 5
my_list = list(map(lambda x: x**2, [1, 2, 3, 4, 5]))

print(my_list)
```

Результат [1, 4, 9, 16, 25]

**filter()** - Как следует из названия, **filter()** фильтрует последовательность по заданному условию.

```
# отфильтруем список с целью получить только чётные значения
my_list = list(filter(lambda x: x % 2 == 0, [11, 22, 33, 44, 55, 66]))

print(my_list)
```

Результат [22, 44, 66]

1. Напишите программу Python для создания лямбда-функции, которая добавляет 15 к заданному числу, переданному в качестве аргумента, а также создайте лямбда-функцию, которая умножает аргумент **x** на аргумент **y** и распечатывает результат.

Решение

```
r = lambda a : a + 15
print(r(10))
```

```
r = lambda x, y : x * y
print(r(12, 4))
```

2. Напишите программу Python для создания функции, которая принимает один аргумент, и этот аргумент будет умножен на неизвестное заданное число.

Решение

```
def func_compute(n):
    return lambda x : x * n
result = func_compute(2)
print("Double the number of 15 =", result(15))
result = func_compute(3)
print("Triple the number of 15 =", result(15))
result = func_compute(4)
print("Quadruple the number of 15 =", result(15))
result = func_compute(5)
print("Quintuple the number 15 =", result(15))
```

3. Напишите программу Python для сортировки списка кортежей с помощью Lambda - функции.

Решение

```
subject_marks = [('English', 88), ('Science', 90), ('Maths', 97), ('Social sciences', 82)]
print("Original list of tuples:")
print(subject_marks)
subject_marks.sort(key = lambda x: x[1])
print("\nSorting the List of Tuples:")
print(subject_marks)
```

Во второй строке через пробел задаются N чисел – элементы массива. Программа должна вывести в одной строке номера элементов массива, равных минимальному, разделив их пробелами:

```
print(*(lambda _, a: (index + 1 for index, value in enumerate(a) if value == min(a))) (int(input()), list(map(int, input().split()))))
print(*sorted(a, key=lambda x: x % 10))
print(*(lambda _, a: (index + 1 for index, value in enumerate(a) if value == min(a))) (int(input()), list(map(int, input().split()))))
```

## Тест-задание 14. Лямбда-функции

### Вопрос 1

Напишите программу, которая сортирует элементы массива по возрастанию последней цифры десятичной записи чисел. Используя lambda- функции.

#### Входные данные

Первая строка содержит размер массива  $N$ . Во второй строке через пробел задаются  $N$  чисел – элементы массива. Гарантируется, что  $0 < N \leq 1000$ .

#### Выходные данные

Программа должна вывести в одной строке элементы массива, отсортированного в порядке возрастания последней цифры в десятичной записи чисел, разделив их пробелами. Числа, у которых последняя цифра одинаковая, должны быть выведены в том же порядке, в котором они стояли в исходной последовательности.

**For example:**

Input	Result
6 219 234 890 81 73 96	6 219 234 890 81 73 96 890 81 73 234 96 219

**Answer:(penalty regime: 10, 20, ... %)**

<pre>n = int(input()) a = map(int, input().split(maxsplit = n)) print(*sorted(a, key=lambda x: x % 10))</pre>	<pre>m=map(int,input().split()) a=list(map(int,input().split())) print(*sorted(a,key=lambda n: n % 10))</pre>
---	---

### Вопрос 2

Напишите программу, которая сортирует натуральные числа в массиве по убыванию суммы цифр десятичной записи числа. При равенстве сумм цифр числа должны сохранить исходный порядок. Использовать lambda-функции.

#### Входные данные

Первая строка содержит размер массива  $N$ . Во второй строке через пробел задаются  $N$  чисел – элементы массива. Гарантируется, что  $0 < N \leq 1000$ .

#### Выходные данные

Программа должна вывести в одной строке элементы массива, отсортированного в порядке убыванию суммы цифр десятичной записи числа, разделив их пробелами.

**For example:**

Input	Result
6 9 21 32 55 81 11	6 9 21 32 55 81 11 55 9 81 32 21 11

**Answer:(penalty regime: 10, 20, ... %)**

**НЕВЕРНО**

**ВЕРНО**

<pre> n = int(input()) a = input().split(maxsplit = n) print(*a.sort(key = lambda x: -sum(map(int, x)))) </pre>	<pre> m=map(int,input().split()) a=input().split() print(*sorted(a, reverse=True, key=lambda x: sum(int (i) for i in str(x)))) </pre>
<pre> n = int(input()) a = input().split(maxsplit = n) print(*a.sort(key = lambda x: sum(map(int, x)), reverse = True)) </pre>	
<pre> n = int(input()) a=input().split(maxsplit = n) print(*a.sort(key=lambda x: sum(int (i) for i in str(x)), reverse = True)) -- </pre> <div> <p><b>Здесь ошибка:</b> ***Error***</p> <pre> Traceback (most recent call last):   File "prog.python3", line 9, in     print(*a.sort(key=lambda x: sum(int (i) for i in str(x)), reverse = True)) TypeError: print() argument after * must be an iterable, not NoneType </pre> </div>	



## Программный тест 9. Модули и пакеты

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1) Напишите программу на языке Python для вывода информации по следующим системным характеристикам: номер используемой версии Python, номер версии интерпретатора Python, номер сборки и наименование используемого компилятора.

Решение:

```
import sys
print("Python version")
print(sys.version)
print("Version info.")
print(sys.version_info)
```

2) Напишите программу для вывода информации по текущему имени пользователя.

Решение:

```
import getpass
print(getpass.getuser())
```

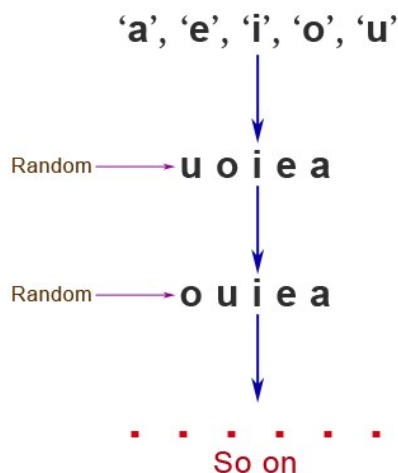
3) Напишите программу для сортировки целых чисел, вводимых с клавиатуры, без использования условных операторов и циклов.



Решение:

```
x = int(input("Input first number: "))
y = int(input("Input second number: "))
z = int(input("Input third number: "))
a1 = min(x, y, z)
a3 = max(x, y, z)
a2 = (x + y + z) - a1 - a3
print("Numbers in sorted order: ", a1, a2, a3)
```

4) Напишите программу Python для создания строки случайным образом, используя 'a', 'e', 'i', 'o', 'u'. Используйте символы ровно один раз.



#### Решение

```
import random
char_list = ['a','e','i','o','u']
random.shuffle(char_list)
print(''.join(char_list))
```

5) Напишите программу для отображения различных форматов даты и времени. Примечание используйте модуль DateTime

- a) Текущая дата и время
- b) Текущий год
- c) Месяц года
- d) Номер недели в году
- e) День недели
- f) День года
- g) День месяца
- h) День недели

#### Решение

```
print("Current date and time: " , datetime.datetime.now())
print("Current year: ", datetime.date.today().strftime("%Y"))
print("Month of year: ", datetime.date.today().strftime("%B"))
print("Week number of the year: ", datetime.date.today().strftime("%W"))
print("Weekday of the week: ", datetime.date.today().strftime("%w"))
print("Day of year: ", datetime.date.today().strftime("%j"))
print("Day of the month : ", datetime.date.today().strftime("%d"))
print("Day of week: ", datetime.date.today().strftime("%A"))
```

## Тест-задание 9. Методы и пакеты

### Вопрос 1

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

Answer:(penalty regime: 10, 20, ... %)

неверно???

Верно

```
from math import tan, pi
n_sides = int(input("Input number of sides: "))
s_length = float(input("Input the length of a side: "))
p_area = n_sides * (s_length ** 2) / (4 * tan(pi / n_sides))
print("The area of the polygon is: ",p_area)
```

```
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)
```

### Вопрос 2

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

Answer:(penalty regime: 10, 20, ... %)

```
from decimal import *
x = Decimal('1.44')
print("Square root of ",x, " is :", x.sqrt())
print("exponential of ",x, " is :", x.exp())
```

### Вопрос 3

Напишите программу Python для преобразования полярных координат в прямоугольные координаты .

Answer:(penalty regime: 10, 20, ... %)

```
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)
```

### Вопрос 4

Реализовать программу для преобразования числа с плавающей запятой в соотношение.

Answer:(penalty regime: 10, 20, ... %)

```
import time
from fractions import Fraction
```

```
value = 4.2
print(Fraction(value).limit_denominator())
```

### Вопрос 5

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

For example:

Result
$2/3 + 3/7 = 23/21$
$2/3 - 3/7 = 5/21$
$2/3 * 3/7 = 2/7$
$2/3 / 3/7 = 14/9$

Answer:(penalty regime: 10, 20, ... %)

неверно???

Верно

<pre>import fractions f1 = fractions.Fraction(2, 3) f2 = fractions.Fraction(3, 7) print('{} + {} = {}'.format(f1, f2, f1 + f2)) print('{} - {} = {}'.format(f1, f2, f1 - f2))</pre>	<pre>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))</pre>
---	---

### Вопрос 6

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

Answer:(penalty regime: 10, 20, ... %)

```
from math import pi
print(pi)
```

## Задание. Модули и пакеты

Здравствуйте, уважаемые слушатели!

Для усвоения темы, Вам необходимо выполнить практическое задание по пройденному материалу.

- Откройте среду разработки Wing 101. Изучите примеры решения задач, разобранных в [ВИДЕОЛЕКЦИЯ. Практическая часть. Модули и пакеты](#).
- Создайте файл для каждой новой задачи.

**Требования к оформлению:** после выполнения задания сохраните файл. По умолчанию он будет сохранен в формате Python с расширением «.py» Данный файл необходимо назвать: *Т6, (номер задачи) и Ваши ФИО*. Далее нажать "**Добавить ответ на задание**" и прикрепить файл.

### Заполнение списков с помощью random. Генерация случайных объектов в Python

Для использования случайных объектов используется модуль **random**

**from random import \***

Несколько полезных методов модуля

**X = random()** - случайное дробное число от 0 до 1.

**X = randint(A, B)** - случайное целое число N,  $A \leq N \leq B$ .

**X = randrange(A, B, step)** – случайное целое число N,  $A \leq N \leq B$ , с шагом step.

**X = choice(sequence)** - случайный элемент непустой последовательности.

**shuffle(sequence)** - перемешивает последовательность (изменяется сама последовательность). Поэтому функция не работает для неизменяемых объектов.

**sample(population, k)** - список длиной k из последовательности population.

#### Примеры:

<code>a = [random() for i in range(10)]</code>	Создается список из 10 дробных чисел от 0 до 1 не включительно
<code>b = [randint(1, 10) for i in range(10)]</code>	Создается список из 10 целых чисел от 1 до 10 включительно
<code>shuffle(a)</code>	Элементы списка <b>a</b> перемешиваются
<code>b = sample(a, 5)</code>	Создается выборка из 5 элементов из списка <b>a</b>
<code>x = choice(['red', 'green', 'blue'])</code> <code>cp = choice('red')</code>	Выбирается один случайный элемент из списка

#### Примеры

##### Задание 1. `from random import *`

```
n = 10
a = [randint(1, 10) for i in range(n)]
print(a)
```

Выходные данные:

[3, 5, 9, 5, 4, 6, 1, 1, 7, 8]

##### Задание 2. `from random import *`

```
a = [round(random(),2) for i in range(5)]
```

```
b = [randint(1, 5) for i in range(5)]
print(a)
print(b)
```

*Выходные данные:*

[0.55, 0.11, 0.27, 0.99, 0.41]

[4, 2, 5, 5, 4]

**Задание 3.** from random import\*

```
a = [round(random(),3) for i in range(5)]
b = [randint(1, 5) for i in range(5)]
shuffle(a)
b = sample(a, 5)
x = choice(['red', 'green', 'blue'])
print(a)
print(b)
print(x)
```

*Выходные данные:*

[0.943, 0.165, 0.526, 0.243, 0.403]

[0.943, 0.403, 0.526, 0.243, 0.165]

Red

**Задание 4.** Напишите программу, которая вводит два целых числа,  $a$  и  $b$  ( $a < b$ ), и выводит через пробел 5 случайных целых чисел на отрезке  $[a, b]$ .

*Входные данные*

Входная строка содержит два целых числа  $a$  и  $b$ , разделённых пробелом. Гарантируется, что  $a < b$ .

*Выходные данные*

Нужно вывести в одной строке через пробел 5 случайных целых чисел на отрезке  $[a, b]$ .

**Примеры**

входные данные

0 100

выходные данные

0 3 86 20 27

**Решение**

```
a, b = map(int, input().split())
from random import randint
c = randint(a,b)
d = randint(a,b)
e = randint(a,b)
g = randint(a,b)
f = randint(a,b)
print(c, ' ', d, ' ', e, ' ', g, ' ', f)
```

**Задание 5.** Напишите программу, которая вводит два вещественных числа,  $a$  и  $b$  ( $a < b$ ), и выводит через пробел 5 случайных вещественных чисел в полуинтервале  $[a, b)$ .

Входные данные

Входная строка содержит два вещественных числа  $a$  и  $b$ , разделённых пробелом. Гарантируется, что  $a < b$ .

#### Выходные данные

Нужно вывести в одной строке через пробел 5 случайных вещественных чисел в полуинтервале  $[a, b)$  с тремя знаками в дробной части.

#### **Примеры**

##### Входные данные

1.234 5.678

##### Выходные данные

1.615 5.543 3.128 2.142 2.290

#### **Решение:**

```
a,b = map(float, input().split())
from random import uniform
c = uniform(a,b)
d = uniform(a,b)
e = uniform(a,b)
g = uniform(a,b)
f = uniform(a,b)

print("{:6.3f}".format(c), "{:6.3f}".format(d), "{:6.3f}".format(e), "{:6.3f}".format(g), "{:6.3f}".format(f))
```

#### **Задачи для самостоятельных решений**

1. Сгенерируйте  $N$  целых чисел от -35 до 35.

Решение:

```
import random
n = int(input())
mas = [random.randint(-35,35) for i in range (n)]
print(*mas)
```

2. Сгенерируйте  $N$  двухзначных чисел.

Решение:

```
import random
n = int(input())
mas = [random.randint(10,99) for i in range (n)]
print(*mas)
```

3. Сгенерируйте  $N$  пар случайных натуральных чисел не более 10000, так что 1-е число нечетное, второе – четное.

Пример для  $N = 3$

3 78
765 20
13 888

Решение:

```
import random
n = int(input())
g=0
```

```

while g <= n:
    k=0
    l=3
    while k % 2 == 0:
        k= random.randint(0,10000)
    while l % 2 != 0:
        l= random.randint(0,10000)
    print(k, l)
    g+=1

```

4. Сгенерируйте случайную перестановку из N чисел (числа от 0 до N-1)

Пример:

5
4 3 2 0 1

Решение

```

import random
n = int(input())
arr = [i for i in range(n)]
arr_s=sorted(arr, reverse=True)
k= random.randint(0,n-1)
l= random.randint(0,n-1)
#arr[k], arr[l] = arr[l], arr[k]
#print(*arr)
arr_s[k], arr_s[l] = arr_s[l], arr_s[k]
print(*arr_s)

```

5. Сгенерируйте N дробных целых чисел от -100 до 100 с 2 знаками после запятой.

Пример:

5
78.97 24.99 -5.92 -99.01

Решение

```

import random
n = int(input())
a = [random.random()*(100+100)-100 for i in range(-100, 100)]
#print(*a, sep=' ')
for i in range(len(a)):
    a[i] = round(a[i], 2)
print(*a)

```

6. Сгенерируйте строку из N заглавных букв английского алфавита

Пример:

107
HPBTVCAPLUFCLBWQOPLZICENJLGOGONUHBUDAZOJEDZEGPWQJHCWVOTDYCIDBKXRLN YGMCPUTEIOPEIBSGCOPCTIPAWMISEAPLIQIDMI



Решение

```
import random
n = int(input())
data='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
data_l=list(data)
mas = [random.choice(data_l) for i in range (n)]
print(*mas, sep = '')
```

7. Сгенерируйте пароль – строку из 12 символов, содержащую заглавные буквы английского алфавита, строчные и цифры.

Пример:

12 wUt2Bs88jKW H
------------------------

Решение:

```
import random
data='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'
data_l=list(data)
mas = [random.choice(data_l) for i in range (12)]
print(*mas, sep = '')
```

## Программный тест по теме. Регулярные выражения

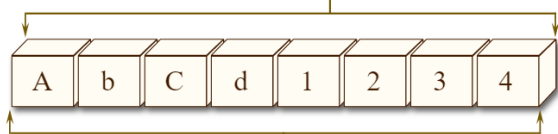
Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1) Напишите программу Python, чтобы проверить, что строка содержит только набор символов a-z, A-Z и 0-9).



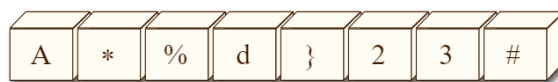
Checking the string for contains  
certain set of characters

Checking a-z, A-Z and 0-9



True

© w3resource.com



Checking the string for contains  
certain set of characters

Checking a-z, A-Z and 0-9



These are not including in the set of characters

False

© w3resource.com

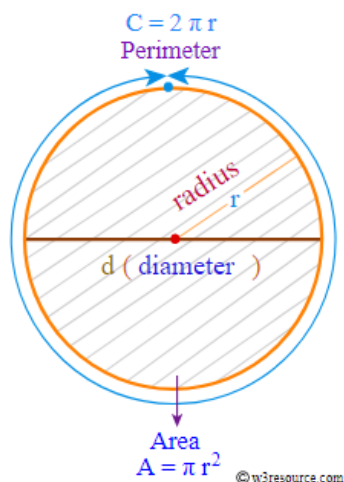
```
import re
#функция проверки
def is_allowed_specific_char(string):
    charRe = re.compile(r'^a-zA-Z0-9.$') # задание выражения
    string = charRe.search(string) # вызов функции поиска в строке заданного
    выражения
    return not bool(string)
print(is_allowed_specific_char(input()))
```

## Тест-задание. Регулярные выражения

## Программный тест по теме Объектно-ориентированное программирование

Прежде чем начать тестирование, ознакомьтесь с примерами решений ниже.

1) Напишите класс Python с именем Circle, построенный с помощью радиуса и двух методов, которые будут вычислять площадь и длину окружности.



Решение

```
class Circle():
    def __init__(self, r):
        self.radius = r
    def area(self):
        return self.radius**2*3.14
    def perimeter(self):
        return 2*self.radius*3.14
NewCircle = Circle(8)
print(NewCircle.area())
print(NewCircle.perimeter())
```

2) Пример реализации класса "Преобразования целого числа в римское число"

```
class py_solution:
    def int_to_Roman(self, num):
        val = [
            1000, 900, 500, 400,
            100, 90, 50, 40,
            10, 9, 5, 4,
            1
        ]
        syb = [
            "M", "CM", "D", "CD",
            "C", "XC", "L", "XL",
            "X", "IX", "V", "IV",
            "I"
        ]
```

```
    ]
    roman_num = ""
    i = 0
    while num > 0:
        for _ in range(num // val[i]):
            roman_num += syb[i]
            num -= val[i]
        i += 1
    return roman_num

print(py_solution().int_to_Roman(1))
print(py_solution().int_to_Roman(4000))
```

