



Множество, Кортеж, Словарь

Кортеж. тип данных tuple

Списки – изменяемые коллекции, **строки** – неизменяемые последовательности Unicode символов. В Python имеются и неизменяемые последовательности, содержащие, в отличие от строк, абсолютно произвольные данные. Такие коллекции называются **кортежами**.

Списки могут делать то же, что кортежи, и даже больше. Но неизменяемость **кортежей** обеспечивает им особые свойства:

- **скорость** – кортежи быстрее работают, так как из-за неизменяемости хранятся в памяти иначе, и операции с их элементами выполняются заведомо быстрее, чем с компонентами списка. Одна из причин существования кортежей – производительность. Обработка кортежа выполняется быстрее, чем обработка списка, поэтому кортежи удобны для обработки большого объема неизменяемых данных.
- **безопасность** – неизменяемость превращает их в идеальные константы. Заданные кортежами константы делают код более читаемым и безопасным. Кроме того, в кортеже можно безопасно хранить данные, не опасаясь, что они будут случайно или преднамеренно изменены в программе.

Кортеж. тип данных tuple

```
empty_tuple = ()           # пустой кортеж
point = (1.5, 6.0)         # кортеж из двух чисел
names = ('Timur', 'Ruslan', 'Roman') # кортеж из трех строк
info = ('Timur', 'Guev', 28, 170, 60, False) # кортеж из 6 элементов разных типов
nested_tuple = (('one', 'two'), ['three', 'four']) # кортеж из кортежа и списка
```

Списки предназначены для объединения неопределенного количества однородных сущностей. Кортежи, как правило, объединяют под одним именем несколько разнородных объектов, имеющих различный смысл. Например, список удобен для хранения нескольких городов. А кортеж удобен для хранения данных о людях.

```
my_tuple = (1, 'python', [1, 2, 3])
print(my_tuple)
my_tuple[2][0] = 100
my_tuple[2].append(17)
print(my_tuple)
```

Тот факт, что кортеж является неизменяемым вовсе не означает, что мы не можем поменять содержимое списка в кортеже. При этом важно понимать: меняется список, а не кортеж. Списки являются ссылочными типами данных, поэтому в кортеже хранится ссылка на список, которая не меняется при изменении самого списка.

Кортеж. тип данных tuple

Кортежи поддерживают те же операции, что и списки, за исключением изменяющего содержимое.

Кортежи поддерживают:

- доступ к элементу по индексу (только для получения значений элементов);
- методы, в частности `index()`, `count()`;
- встроенные функции, в частности `len()`, `sum()`, `min()` и `max()`;
- срезы;
- оператор принадлежности `in`;
- операторы конкатенации (+) и повторения (*)
- Подобно спискам, мы можем создавать вложенные кортежи.

Кортеж. тип данных tuple

1. В переменную `city_name` вводится название города (например, Москва), а в переменную `city_year` – год его основания (например, 1147). Получите кортеж из значений этих двух переменных (сначала название города, затем год основания).
2. Получите список, содержащий только непустые кортежи исходного списка `tuples`, не меняя порядка их следования.

Пример входного списка: `tuples = [(), (), ('',), ('a', 'b'), (), ('a', 'b', 'c'), (1,), (), (), ('d',), ('', ''), ()]`

3. Программист Тимур написал программу для работы с биографическими данными русских поэтов. Данные содержатся в кортежах вида (фамилия, год рождения, город рождения). В процессе работы программы в некотором кортеже `poet_data` обнаружилась ошибка: ('Пушкин', 1799, 'Санкт-Петербург'). Тут неверно указано место рождения, ведь Александр Пушкин родился в Москве.

Дополните приведенный ниже код так, чтобы в переменной `poet_data` находился правильный кортеж (с исправленным значением), а затем выведите его.

```
poet_data = ('Пушкин', 1799, 'Санкт-Петербург')
```

Кортеж. тип данных tuple

4. Дополните приведенный ниже код так, чтобы он вывел список, содержащий средние арифметические значения чисел каждого вложенного кортежа в заданном кортеже кортежей `numbers`.

```
numbers = ((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4), (90, 10))
```

5. Напишите программу, которая считывает натуральное число n и выводит первые n чисел последовательности Трибоначчи.

Формат входных данных

На вход программе подается натуральное число n ($n \leq 100$) – количество членов последовательности.

Формат выходных данных

Программа должна вывести члены последовательности Трибоначчи, отделенные символом пробела.

Примечание. Последовательность Трибоначчи – последовательность натуральных чисел, где каждое последующее число является суммой трех предыдущих:

1, 1, 1, 3, 5, 9, 17, 31, 57, 105, ...

Пример: С использованием кортежей многие алгоритмы приобретают достаточно краткую форму. Например, вычисление чисел Фибоначчи может выглядеть следующим образом:

```
n = int(input())
```

```
f1, f2 = 1, 1
```

```
for i in range(n):
```

```
    print(f1)
```

```
    f1, f2 = f2, f1 + f2
```

Множества. Тип данных set

Этот тип данных аналогичен математическим множествам, он поддерживает быстрые операции проверки наличия элемента в множестве, добавления и удаления элементов, операции объединения, пересечения и многие другие.

Важно:

- все элементы множества различны (уникальны), два элемента не могут иметь одинаковое значение;
- множества неупорядочены, то есть элементы не хранятся в каком-то определенном порядке;
- элементы множества должны относиться к неизменяемым типам данных;
- хранящиеся в множестве элементы могут иметь разные типы данных.

Множества. Тип данных set

1. Дан список чисел, который могут содержать до 100000 чисел каждый. Определите, сколько в нем встречается различных чисел.
2. Даны два списка чисел, которые могут содержать до 100000 чисел каждый. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
3. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.
4. На вход программе подается строка текста. Напишите программу, которая определяет количество различных символов в строке.
5. На вход программе подаются две строки, состоящие из цифр. Необходимо определить, верно ли, что для записи этих строк были использованы одинаковые наборы цифр?
6. На вход программе подается строка, состоящая из трех слов. Верно ли, что для записи всех трех слов был использован один и тот же набор букв?

Множества. Тип данных set

1. Дан список чисел, который могут содержать до 100000 чисел каждый. Определите, сколько в нем встречается различных чисел.
2. Даны два списка чисел, которые могут содержать до 100000 чисел каждый. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
3. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.
4. На вход программе подается строка текста. Напишите программу, которая определяет количество различных символов в строке.
5. На вход программе подаются две строки, состоящие из цифр. Необходимо определить, верно ли, что для записи этих строк были использованы одинаковые наборы цифр?
6. На вход программе подается строка, состоящая из трех слов. Верно ли, что для записи всех трех слов был использован один и тот же набор букв?

Аня и Боря любят играть в разноцветные кубики, причем у каждого из них свой набор и в каждом наборе все кубики различны по цвету. Однажды дети заинтересовались, сколько существуют цветов таких, что кубики каждого цвета присутствуют в обоих наборах. Для этого они занумеровали все цвета случайными числами. На этом их энтузиазм иссяк, поэтому вам предлагается помочь им в оставшейся части.

Номер любого цвета — это целое число в пределах от 0 до 10^9 . В первой строке входного файла записаны числа N и M — количество кубиков у Ани и Бори соответственно. В следующих N строках заданы номера цветов кубиков Ани. В последних M строках номера цветов кубиков Бори.

Выведите сначала количество, а затем отсортированные по возрастанию номера цветов таких, что кубики каждого цвета есть в обоих наборах, затем количество и отсортированные по возрастанию номера остальных цветов у Ани, потом количество и отсортированные по возрастанию номера остальных цветов у Бори.

Ввод	Вывод
4 3	2
0	0 1
1	2
10	9 10
9	1
1	3
3	
0	

Август и Беатриса играют в игру. Август загадал натуральное число от 1 до n . Беатриса пытается угадать это число, для этого она называет некоторые множества натуральных чисел. Август отвечает Беатрисе **YES**, если среди названных ей чисел есть задуманное или **NO** в противном случае. После нескольких заданных вопросов Беатриса запуталась в том, какие вопросы она задавала и какие ответы получила и просит вас помочь ей определить, какие числа мог задумать Август.

Первая строка входных данных содержит число n — наибольшее число, которое мог загадать Август. Далее идут строки, содержащие вопросы Беатрисы. Каждая строка представляет собой набор чисел, разделенных пробелами. После каждой строки с вопросом идет ответ Августа: **YES** или **NO**.

Наконец, последняя строка входных данных содержит одно слово **HELP**. Вы должны вывести (через пробел, в порядке возрастания) все числа, которые мог задумать Август.

Ввод	Вывод
10 1 2 3 4 5 YES 2 4 6 8 10 NO HELP	1 3 5