

Библиотека `itertools` в Python

Что такое itertools?

- ▶ `itertools` — это встроенная библиотека Python, содержащая набор инструментов для работы с итераторами.
- ▶ Зачем нужна? Позволяет легко и эффективно генерировать сложные последовательности: комбинации, перестановки, декартовы произведения и т.д.
- ▶ Преимущества:
 - ▶ Экономит время на написании кода перебора.
 - ▶ Код становится короче, чище и понятнее.
 - ▶ Работает очень быстро (реализована на языке C).

```
from itertools import product, permutations
# или
import itertools
```

Метод `product` (Декартово произведение)

Что делает? Генерирует все возможные комбинации элементов из переданных последовательностей.

Аналог в математике: Декартово произведение множеств.

Аналог в программировании: Вложенные циклы `for`.

Синтаксис:

```
product(iterable1, iterable2, ..., repeat=1)
```

`repeat` — необязательный аргумент. Если нужно найти произведение последовательности на саму себя, можно указать `repeat=n` вместо многократной передачи одного и того же аргумента.

The diagram shows the Cartesian product of two sets, A and B, where both sets contain elements A, B, and C. The sets are represented as $\begin{pmatrix} A \\ B \\ C \end{pmatrix}$ and $\begin{pmatrix} A \\ B \\ C \end{pmatrix}$, with a multiplication sign between them. The result is shown as a list of all possible combinations: AA, AB, AC, BA, BB, BC. To the right of each combination, there is a red checkmark, indicating that all combinations are valid results of the Cartesian product.

Примеры

```
from itertools import product
```

```
dice = [1, 2, 3, 4, 5, 6]
```

```
result = list(product(dice, repeat=2)) # Все исходы броска 2 кубиков
```

```
print(len(result)) # 36
```

```
print(result[:5]) # [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5)]
```

Ещё пример

```
from itertools import product
```

```
a = ['x', 'y']
```

```
b = [1, 2]
```

```
result = list(product(a, b))
```

```
print(result)
```

```
# Вывод: [('x', 1), ('x', 2), ('y', 1), ('y', 2)]
```

Решение задачи ЕГЭ с product

- ▶ Алексей составляет 5-буквенные пароли, используя только символы из набора {А, В, С, D, Е, F}. Причем пароль не может начинаться с буквы F и должен содержать хотя бы одну гласную (А или Е). Сколько различных паролей может составить Алексей?

```
from itertools import product
```

```
count = 0
```

```
# Генерируем все возможные 5-буквенные слова
```

```
for p in product('ABCDEF', repeat=5):
```

```
    word = ''.join(p)
```

```
    # Проверяем условия:
```

```
    # 1. Первая буква не 'F'
```

```
    # 2. В слове есть хотя бы одна гласная (А или Е)
```

```
    if word[0] != 'F' and ('A' in word or 'E' in word ):
```

```
        count += 1
```

```
print(count) # Выведет правильный ответ
```

21407

Виктор составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Виктор использует 5-буквенные слова, в которых могут быть только буквы Д, Г, И, А, Ш, Э, причём слово не должно начинаться с гласной и не должно заканчиваться согласной. Сколько различных кодовых слов может использовать Виктор?

18963

Александр составляет 8-буквенные слова, в которых могут быть использованы только буквы К, О, Т, Б, У, С. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Слово должно содержать в себе подстроку КОТ и не должно начинаться с гласной буквы. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Александр?

20270

Сколько существует пятизначных семеричных **чисел**, в которых не менее 2-х раз четные цифры стоят рядом, при этом **всем** никакие три четные цифры не стоят рядом.

В ответе укажите только число.

23192

Все шестибуквенные слова, составленные из букв Т, Е, О, Р, И, Я, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. EEEEEЕ
2. EEEEEИ
3. EEEEEО
4. EEEEEР
5. EEEEEТ
6. EEEEEЯ

Определите, под каким номером в этом списке стоит последнее слово с нечётным номером, которое не начинается с букв Р, Т или Я и при этом содержит в своей записи не менее двух букв И.

Примечание. Слово - последовательность идущих подряд букв, не обязательно осмысленная.

23554

Все пятибуквенные слова, составленные из букв А, Л, Г, О, Р, И, Т, М, записаны в алфавитном порядке и пронумерованы.
Вот начало списка:

1. ААААА
2. ААААГ
3. ААААИ
4. ААААЛ
5. ААААМ
6. ААААО
7. ААААР

Определите, под каким номером в этом списке стоит первое слово с чётным номером, которое не начинается с букв А или Г и при этом содержит в своей записи не менее двух букв Р.

Примечание. Слово - последовательность идущих подряд букв, не обязательно осмысленная.

21703

Все шестибуквенные слова, в составе которых могут быть только буквы П, О, Б, Е, Д, А, записаны в алфавитном порядке и пронумерованы начиная с 1.

Ниже приведено начало списка.

1. АААААА
2. АААААБ
3. АААААД
4. АААААЕ
5. АААААО
6. АААААП

...

Определите последний чётный номер слова, которое начинается с буквы О и в котором каждая буква встречается ровно один раз.

Примечание. Слово - последовательность идущих подряд букв, не обязательно осмысленная.

ФИПИ

Миша заполнял таблицу истинности логической функции F

$$(x \vee \neg y) \wedge \neg(y \equiv z) \wedge \neg w,$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
1	1			1
	1	0	0	1
1		1	0	1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т. д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция F задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		F
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму столбцу – переменная x . В ответе следует написать: yx .

```
from itertools import product
```

```
print("w x y z F")
```

```
for w, x, y, z in product([0, 1], repeat=4):
```

```
    F = ...
```

```
    if F == ...:
```

```
        print(w, x, y, z, ...)
```

23739

Миша заполнял таблицу истинности логической функции $F = (x \vee y) \wedge \neg(y \equiv z) \wedge \neg w$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
1		1		1
0	1		0	1
	1	1	0	1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

23261

Миша заполнял таблицу истинности логической функции $F = \neg(w \rightarrow (x \equiv y)) \wedge (z \rightarrow x)$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
	0	1	0	1
0			0	1
	1	1		1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

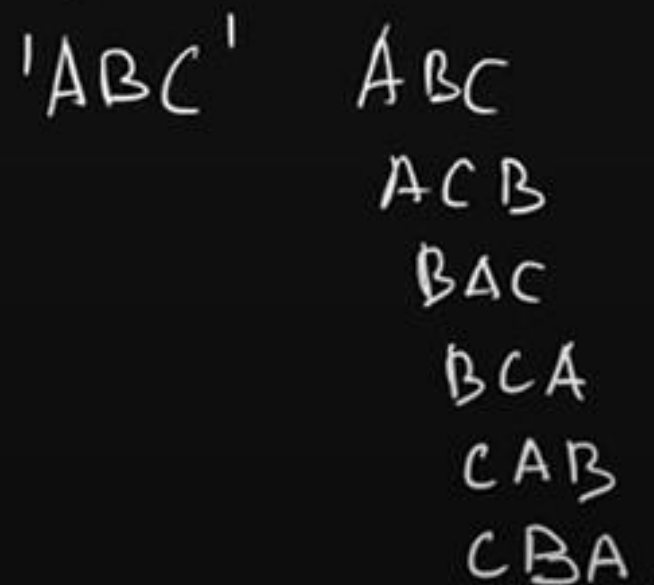
Метод `permutations` (Перестановки)

- ▶ **Что делает?** Генерирует все возможные перестановки элементов входной последовательности.
- ▶ **Порядок элементов важен!** (1, 2, 3) и (3, 2, 1) — это разные перестановки.
- ▶ **Длина по умолчанию:** равна длине последовательности. Можно задать свою длину (`r`).

`permutations(iterable, r=None)`

`r` — длина возвращаемых перестановок.

Если `r` не указана, то `r = len(iterable)`



'ABC' ABC
ACB
BAC
BCA
CAB
CBA

Пример

```
data = ['a', 'b', 'c']  
result = list(permutations(data))  
print(result)
```

Пример

```
from itertools import permutations
```

```
data = [1, 2, 3, 4]
```

```
result = list(permutations(data, r=2)) # Все двухсимвольные коды из 4 цифр
```

```
print(result)
```

Решение задачи ЕГЭ с permutations

Маша составляет 5-буквенные коды из букв В, У, А, Л, Ъ. Каждую букву нужно использовать ровно 1 раз, при этом буква Ъ не может стоять на первом месте и перед гласной. Сколько различных кодов может составить Маша?

```
from itertools import permutations
n = 0
for x in permutations('ВУАЛб'):
    s = ''.join(x)
    if s[0] != 'б' and 'бУ' not in s \
        and 'бА' not in s:
        n += 1
print(n)
```

Множество (set)

- ▶ **Что это:** Изменяемая неупорядоченная коллекция **уникальных** элементов
- ▶ **Основные характеристики:**
 - ▶ Элементы не повторяются
 - ▶ Нет порядка элементов
 - ▶ Можно изменять (добавлять/удалять элементы)
 - ▶ Работает очень быстро для проверки принадлежности

```
my_set = {1, 2, 3, 2, 1}    # {1, 2, 3}
```

```
my_set = set([1, 2, 3, 2, 1]) # {1, 2, 3}
```

Методы:

```
s = {1, 2, 3}
```

```
s.add(4)      # {1, 2, 3, 4}
```

```
s.remove(2)   # {1, 3, 4}
```

```
s.discard(5)  # нет ошибки если элемента нет
```

Применение:

Удаление дубликатов

Проверка принадлежности (`x in set`)

Математические операции над множествами

Frozenset

Что это: Неизменяемая версия множества

Основные характеристики:

- ▶ Все свойства обычного множества
- ▶ Нельзя изменить после создания
- ▶ Можно использовать как ключ словаря

Создание:

```
fz = frozenset([1, 2, 3, 2, 1]) # frozenset({1, 2, 3})
```

Особенности:

Не работает - frozenset неизменяем!

```
fz.add(4) # Ошибка!
```

```
fz.remove(1) # Ошибка!
```

Но можно использовать в словарях

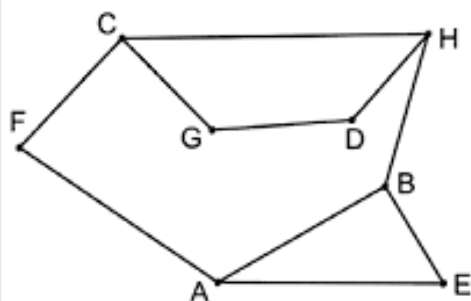
```
d = {frozenset([1, 2]): "value"}
```

Маша составляет шестибуквенные слова перестановкой букв слова КАПКАН. При этом она избегает слов с двумя подряд одинаковыми буквами. Сколько различных кодов может составить Маша?

17664

На рисунке схема дорог N-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

		Номер пункта							
		1	2	3	4	5	6	7	8
Номер пункта	1		5		2				30
	2	5				39		21	
	3				53	13	8		
	4	2		53			1		
	5		39	13					
	6			8	1				
	7		21						3
	8	30						3	

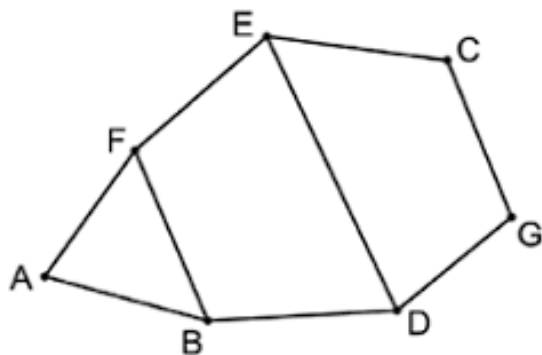


Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённости дорог из пункта C в пункт G и из пункта H в пункт B. В ответе запишите целое число.

17620

На рисунке схема дорог N-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1		8			5	53	
	2	8		2	39			
	3		2				30	3
	4		39					13
	5	5					21	
	6	53		30		21		
	7			3	13			

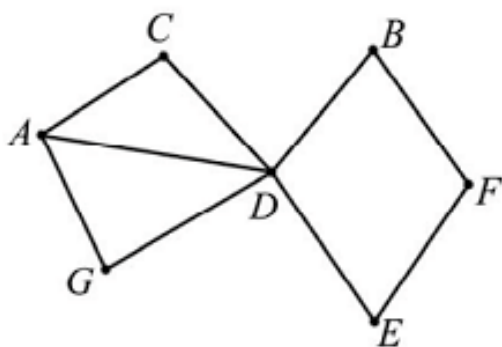


Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённости дорог из пункта F в пункт B и из пункта E в пункт D. В ответе запишите целое число.

16312

На рисунке изображена схема дорог N-ского района, в таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет.

		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1			*				*
	2					*		*
	3	*			*			*
	4			*				*
	5		*				*	
	6					*		*
	7	*	*	*	*		*	



Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам В и Е на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания

Словарь (dict)

Что это: Изменяемая неупорядоченная коллекция пар **ключ-значение**

Основные характеристики:

- ▶ Ключи уникальные и неизменяемые (строки, числа, кортежи, frozenset)
- ▶ Значения любые и могут повторяться
- ▶ Быстрый поиск по ключу (как в хеш-таблице)

Создание:

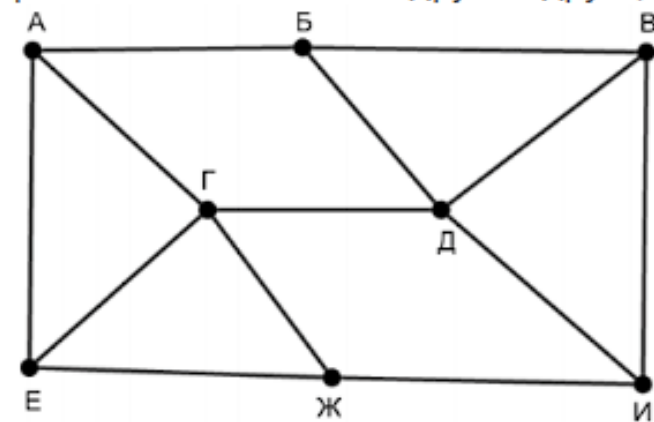
```
d1 = {'a': 1, 'b': 2, 'c': 3}
```

```
d2 = dict(a=1, b=2, c=3)
```

```
d3 = dict([('a', 1), ('b', 2)])
```

1234

На рисунке схема дорог изображена в виде графа, в таблице содержатся сведения о длине этих дорог в километрах. Так как таблицу и схему рисовали независимо друг от друга, нумерация населенных пунктов в таблице никак не связана с буквенными обозначениями на графе.

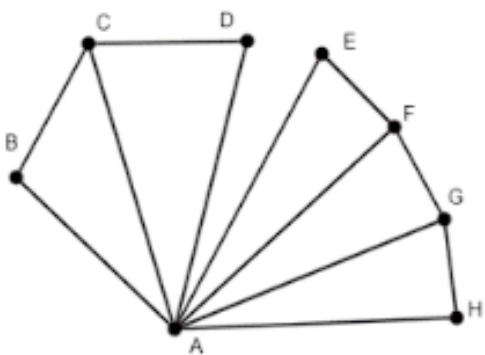


	П1	П2	П3	П4	П5	П6	П7	П8
П1				14	37		25	
П2			18		24			29
П3		18		42	23			21
П4	14		42			20	17	
П5	37	24	23					
П6				20			28	30
П7	25			17		28		
П8		29	21			30		

Известно, что длина дороги ГЕ больше, чем длина дороги ГЖ. Определите длину дороги БВ. В ответе запишите целое число - длину дороги в километрах.

5056

На рисунке схема дорог изображена в виде графа, в таблице содержатся сведения о длине этих дорог в километрах. Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Известно, что дорога CD длиннее дороги EF. Определите сумму длин дорог AB и AG.



	П1	П2	П3	П4	П5	П6	П7	П8
П1		17					32	
П2	17						29	13
П3				16	12		33	
П4			16				28	
П5			12				38	
П6							25	15
П7	32	29	33	28	38	25		30
П8		13				15	30	