

## Практикум по модулю 4

Практика. Перебор списков .....	1
Практика методы списков .....	2
Практика генераторы списков.....	6
Вопросы к тесту .....	8
Ссылка на тест.....	9

### Практика. Перебор списков

1. Дан список. Вывести элементы списка с помощью оператора for.

Каждый элемент должен быть выведен с новой строки.

```
my_list = [24, "Василий", "выхухоль", 42, -12.5]
```

2. Вывести элементы предыдущего списка, пронумеровав их начиная с единицы.

Каждый элемент должен быть выведен с новой строки.

3\*. Сформировать список из n элементов, заполненный произвольными целыми числами в диапазоне от -10 до 10.

Вывести на экран сумму всех элементов данного списка.

4. Вводятся три целых числа в одну строку через пробел. Сформируйте список lst, хранящий эти значения в порядке их ввода. Результат выведите на экран, используя команду:

```
print(lst)
```

Входные данные:

```
8 11 3
```

Выходные данные:

```
[8, 11, 3]
```

5. Вводятся названия городов в одну строчку через пробел. На основе этой строки формируется список с помощью команды:

```
cities = input().split()
```

Необходимо вывести значение последнего элемента этого списка на экран.

---

Входные данные:

Москва Питер Уфа Казань Владимир

---

Выходные данные:

Владимир

### Практика методы списков

1. Даны два целых числа. Вывести список всех чисел кратных трем в диапазоне между заданными числами.

```
first_number = int(input()) # Первое число
```

```
second_number = int(input()) # Второе число
```

2. Дан список из элементов  $x = [-2, -1, 0, 1, 2, 3, 4, 5, 6, 7]$  – это координаты  $x$ , точек, лежащих на прямой, описываемой уравнением  $y = 2x - 4$ .

Получить и вывести на экран список координат  $y$  для данного уравнения прямой.

3\* (дз). Пользователь вводит на экран дату в формате dd.mm.yyyy, например: 02.01.2013.

Ваша задача вывести дату в текстовом виде, например: второе января 2013 года.

Подсказка: создайте списки с названием дней и названиями месяцев.

Примечание: для экономии времени, можно ограничить номер дня десятью.

Примечание: склонениями названий можно пренебречь

4. Вводятся оценки студента (целые числа от 2 до 5) в одну строчку через пробел. На основе введенной строки формируется список командой:

```
marks = list(map(int, input().split()))
```

Необходимо вычислить средний балл и вывести его на экран с точностью до десятых (один знак после запятой).

---

Входные данные:

3 3 2 4 4 5 4 3 2

---

Выходные данные:

3.3

5. Вводится информация по книге (каждое значение с новой строки): название, автор, число страниц (целое число), цена (вещественное число). На основе этих данных формируется список `book` с элементами в порядке их ввода. Затем, из этого списка необходимо удалить 3-й элемент (число страниц), в качестве автора записать "Пушкин" и цену увеличить в 2 раза. Результат вывести на экран командой:

```
print(book)
```

---

Входные данные:

Мастер и Маргарита

Булгаков

233

435.45

---

Выходные данные:

['Мастер и Маргарита', 'Пушкин', 870.9]

6. Вводятся названия городов в одну строку через пробел. На основе этой строки необходимо создать список `lst` и добавить его в конец следующего списка:

```
cities = ["Москва", "Тверь", "Вологда"]
```

Вывести результат на экран командой:

```
print(*lst)
```

Входные данные:

Уфа Казань Севастополь

Выходные данные:

Москва Тверь Вологда Уфа Казань Севастополь

7. Имеется список городов:

```
s = ["Москва", "Ульяновск", "Самара", "Тверь", "Вологда", "Омск", "Уфа"]
```

Необходимо с помощью срезов выбрать из него города через один (начиная с первого) и результат вывести на экран.

Входные данные:

Выходные данные:

['Москва', 'Самара', 'Вологда', 'Уфа']

12. Имеется список городов:

```
c = ["Москва", "Ульяновск", "Самара", "Тверь", "Вологда", "Омск", "Уфа"]
```

Необходимо с помощью срезов выбрать из него города через один (начиная со второго) и результат вывести на экран.

---

Входные данные:

Выходные данные:

['Ульяновск', 'Тверь', 'Омск']

13. Вводится строка с номером телефона в формате:

+7(xxx)xxx-xx-xx

Необходимо преобразовать ее в список `lst` (посимвольно, то есть, элементами списка будут являться отдельные символы строки). Затем, удалить первый '+', число 7 заменить на 8 и убрать дефисы. Отобразить полученный список на экране командой:

```
print("".join(lst))
```

---

Входные данные:

+7(912)123-45-67

---

Выходные данные:

8(912)1234567

14. Вводятся названия рек в одну строчку через пробел. Необходимо все их отсортировать по именам (по возрастанию) и в отсортированном списке удалить первый элемент. Результат отобразить на экране в одну строчку через пробел.

---

Входные данные:

Лена Обь Волга Дон Енисей

---

Выходные данные:

Дон Енисей Лена Обь

15. Строка содержит только заглавные буквы латинского алфавита (ABC...Z). Определите символ, который чаще всего встречается в файле между двумя одинаковыми символами.

Например, в тексте CBCABABACCC есть комбинации CBC, ABA (два раза), BAB и CCC. Чаще всего — 3 раза — между двумя одинаковыми символами стоит B, в ответе для этого случая надо написать B - 3.

Входные данные:

```
FPFHQDFWBPLLIJZPNKJEFVHWNOKPRHUYVDALCXTQXHLSKTJGENYIDHSCP
RBNUBMPICEHRRIQDSWDCWJFNQEPRFLDJFXSBUBECRZXHSPNPCQSACBBG
MZBONQCKFFDAZBKJBTVCHMORBSFTQBHGDNDWXTEUQAHZDZXTOXHGIUKX
MWBOQKTCYNCVPUWISKCNLENHFLQAIQAGNQUOVIFOBMNJOHFTRITCBZTWG
VXSXOJUBZMABSGBUVXBKKPBZPVQHLBLTJVDMIVKWUNDWGNZXOOOWUFTS
NNENKFTFQDTJACANYWCZQDUMBDNYZGCZUIQRIVXINGWJBQTXHQHWALCXIS
XPDJOXYLSQBOFWUETLQJUQHKGFGNZMFIRRDHQTXYENGUVZEKWMDXAWPRN
CPEUORSTADPRDTPQYFTBUHXWJYNESUMMNQFXIYXBCZEPZJXHGBINTXHD
L
```

Выходные данные:

N - 2

16. Вводится матрица чисел из трех строк. В каждой строке числа разделяются пробелом. Необходимо вывести на экран последний столбец этой матрицы в виде строки из трех чисел через пробел.

Входные данные:

```
8 11 12 1
9 4 36 -4
1 12 49 5
```

Выходные данные:

```
1 -4 5
```

17. Имеется вложенный список из трех строк:

```
t = ["Скажи-ка", "дядя", "ведь", "не", "даром"],
    ["Я", "Python", "выучил", "с", "каналом"],
    ["Специалист", "что", "раздавал?"]
```

Необходимо реализовать проверку на наличие в этом списке введенного слова. Результат (True или False) вывести на экран. Решить задачу необходимо без применения условного оператора.

---

Входные данные:

дядя

---

Выходные данные:

True

```
t = ["Скажи-ка", "дядя", "ведь", "не", "даром"],  
     ["Я", "Python", "выучил", "с", "каналом"],  
     ["Специалист", "что", "раздавал?"]]
```

### Практика генераторы списков

1. Вводятся вещественные числа в строку через пробел. Необходимо на их основе сформировать список `lst` с помощью `list comprehension` (генератора списков) из модулей введенных чисел (в списке должны храниться именно числа, а не строки). Результат вывести на экран в виде списка командой:

```
print(lst)
```

---

Входные данные:

5.56 -8.7 1.0 3.14 77.845

---

Выходные данные:

[5.56, 8.7, 1.0, 3.14, 77.845]

2. Вводится семизначное целое положительное число. С помощью `list comprehension` сформировать список `lst`, содержащий цифры этого числа (в списке должны быть записаны числа, а не строки). Результат вывести на экран список командой:

```
print(lst)
```

---

Входные данные:

4567397

---

Выходные данные:

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

3. Вводятся названия городов в строку через пробел. Необходимо сформировать список с помощью `list comprehension`, содержащий названия длиной более пяти символов. Результат вывести в строку через пробел.

---

Входные данные:

Казань Уфа Москва Челябинск Омск Тур Самара

---

Выходные данные:

Казань Москва Челябинск Самара

4. Вводится натуральное число  $n$ . Необходимо сформировать список с помощью `list comprehension`, состоящий из делителей числа  $n$  (включая и само число  $n$ ). Результат вывести на экран в одну строку через пробел.

---

Входные данные:

10

---

Выходные данные:

1 2 5 10

5. Вводится натуральное число  $N$ . Необходимо сгенерировать вложенный список с помощью `list comprehension`, размером  $N \times N$ , где первая строка содержала бы все нули, вторая - все единицы, третья - все двойки и так до  $N$ -й строки. Результат вывести в виде таблицы чисел как показано в примере ниже.

---

Входные данные:

4

---

Выходные данные:

0 0 0 0

1 1 1 1

2 2 2 2

3 3 3 3

5. Вводится список вещественных чисел. С помощью `list comprehension` сформировать список, состоящий из элементов введенного списка, имеющих четные индексы (то есть, выбрать все элементы с четными индексами). Результат вывести на экран в одну строку через пробел.

---

Входные данные:

8.5 11.3 1.0 -4.5 11.34 6.45

---

Выходные данные:

8.5 1.0 11.34

6. Вводятся два списка целых чисел одинаковой длины каждый с новой строки. С помощью `list comprehension` сформировать третий список, состоящий из суммы

соответствующих пар чисел введенных списков. Результат вывести на экран в одну строку через пробел.

---

Входные данные:

1 2 3 4 5  
6 7 8 9 10

---

Выходные данные:

7 9 11 13 15

---

7. Вводится список городов в одной строке. Необходимо с помощью list comprehension сформировать список lst, содержащий вложенные списки из пар:

<город> <численность населения>

Численность населения - целое число в тыс. человек. Вывести результат на экран

---

Входные данные:

Москва 15000 Уфа 1200 Самара 1090 Казань 1300

---

Выходные данные:

[['Москва', 15000], ['Уфа', 1200], ['Самара', 1090], ['Казань', 1300]]

---

Вопросы к тесту

1. К каким типам данных относится список?
2. Какая конструкция или функция позволяют задавать списки?
3. Какой результат сравнения будет получен при выполнении команды:  
[1, 2, 3] > [1, 1, 10]
4. Какой результат сравнения будет получен при выполнении команды:  
[1, 2, 3] < [1, 2, 3, 4]
5. Какой результат сравнения будет получен при выполнении команды:  
[1, 2, 3] >= [1, 2, 3]
6. Какой результат сравнения будет получен при выполнении команды:  
[1, 2, 3, "4"] > [1, 2, 3, "5"]
7. Какой результат сравнения будет получен при выполнении команды:  
[1, 2, 3, 7] > [1, 2, 3, "5"]
8. По какому индексу можно обратиться к значению 5 списка:



```
a = [0, True, "Москва", 5, False, "Омск"]
```

9. Установите соответствия между методами списков и их описанием

append	Добавляет элемент в конец списка
insert	Вставляет элемент в указанное место списка
remove	Удаляет элемент по значению
pop	Удаляет последний элемент, либо элемент с указанным индексом
clear	Очищает список (удаляет все элементы)

10. Имеется вложенный список:

```
a = [True, [1, 0, ["True", ["Истина", "Ложь"], "False"]], False]
```

11. Имеется многомерный список:

```
a = [True, [1, 0, ["True", ["Истина", "Ложь"], "F"]], False]
```

Какую команду следует выполнить, чтобы удалить элемент со значением "F"?

Ссылка на тест

<https://forms.gle/ZvuupRcjJkUPL9rZ8>