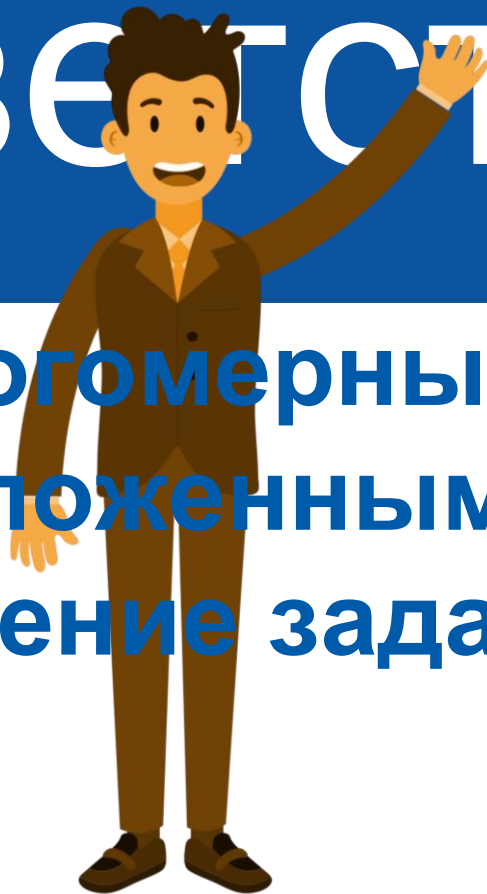


# Приветствие



**Списки. Многомерные списки.  
Циклы по вложенным спискам.  
Решение задач.**

# Повторение

# List.append(x)

Добавляет новый элемент **X** в конец целевого списка. Но не заменяет элементы списка новыми.

# List.insert(index, x)

Добавляет в список элемент **X** по индексу **Index**.

# List.extend(x)

Добавляет набор элементов **X** в конец списка.

# List.remove(x)

Выполняет удаление **X** элемента внутри списка.  
Удаляет только первое вхождение элемента.  
Если не находит **X** генерирует исключение **ValueError**.

# List.pop(x)

Удаляет и возвращает элемент с индексом **X**, при отсутствии аргумента, целевым будет последнее значение в списке.

# List.clear( )

Выполняет удаление всех элементов внутри списка.

# Удаление элементов при помощи `del`

Оператор **`del`** (сокращённое от `delete` - удалить) - позволяет удалять элементы из списков.

# List.index(x)

Возвращает индекс элемента **X**, при отсутствии искомого значения генерируется исключение: Value Error.

# List.count(x)

Считает количество вхождений **X** в список, возвращая результат.

# List.reverse()

Перемещает элементы списка, расставляя их в обратном порядке.

# List.sort( [parameter] )

Сортирует список по заданным параметрам. По умолчанию сортировка ведётся по возрастанию.



# **min( list )**

Возвращает наименьший элемент списка.

# **max( list )**

Возвращает наибольший элемент списка.

# len(list)

Возвращает длину списка.

# sorted(list, [parameter])

Возвращает отсортированный, по установленным параметрам сортировки, список.

# filter(condition, list)

Функция **filter()** позволяет выполнить фильтрацию списка, согласно условия указанного в качестве аргумента **condition**.

# map(function, list)

Функция **map()** позволит выполнить изменение всех элементов списка, согласно условия указанного в качестве аргумента **function**.

# List.copy( ) и copy.copy(x)

Возвращает копию списка. Возвращаемая копия является поверхностной копией целевого списка.

## copy.deepcopy(x)

Возвращает копию списка. Возвращаемая копия является полной копией целевого списка.

# Решение задач

Создать программу, которая генерирует список из  $n$ -чисел от 1 до 4, значение  $n$  вводится вручную. Разделить полученный список на многомерный список значений, строки которого будут содержать одинаковые цифры.



# Результат решения задачи

Сколько чисел генерируем? 10

[4, 1, 3, 4, 2, 4, 1, 2, 4, 2]

[[1, 1], [2, 2, 2], [3], [4, 4, 4, 4]]

Сколько чисел генерируем? 20

[3, 2, 3, 1, 4, 2, 2, 3, 4, 2, 3, 2, 1, 3, 1, 1, 4, 1, 4, 4]

[[1, 1, 1, 1, 1], [2, 2, 2, 2, 2], [3, 3, 3, 3, 3], [4, 4, 4, 4, 4]]

Сколько чисел генерируем? 40

[1, 3, 1, 3, 1, 1, 3, 4, 1, 3, 4, 1, 2, 3, 3, 1, 2, 2, 3, 1, 1, 4, 2, 3, 3, 1, 4, 2, 3, 3, 3, 3, 3, 2, 3, 1, 4, 2, 1, 2]

[[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [2, 2, 2, 2, 2, 2, 2, 2, 2], [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3], [4, 4, 4, 4, 4]]



Программа выполняющая поиск слов, в которых хотя бы один раз, встречаются две одинаковые буквы. Перед началом поиска, производится ввод слов, которые должны быть сохранены в виде списка, ввод осуществляется по одному слову до тех пор, пока не будет введено “пустое” значение. Добавить удаление пробелов из введённых слов. Результаты работы программы сохранить в новый многомерный список.



# Результат решения задачи

Введите слово: Привет

Привет-добавлено в список

Введите слово: Класс

Класс-добавлено в список

Введите слово: До бро

Добро-добавлено в список

Введите слово: в в о д

ввод-добавлено в список

Введите слово: данных

данных-добавлено в список

Введите слово: проверка

проверка-добавлено в список

Введите слово: Значений

Значений-добавлено в список

Введите слово:

```
['Привет', 'Класс', 'Добро', 'ввод', 'данных', 'проверка', 'Значений']
```

```
[['Класс', 'ввод', 'данных'], ['Привет', 'Добро', 'проверка', 'Значений']]
```

Создать программу генерирующую случайный список студентов из заданных данных:

- ☐ Количество студентов.
- ☐ Имя (случайное из списка имён).
- ☐ Фамилия (случайное из списка фамилий).
- ☐ Год поступления (случайный с 2010 до 2023).

Написать функцию определяющую курс, на котором учится студент, в зависимости от года поступления, или год выпуска.

Расчёт курса студента:

- ☐ 1й курс - поступившие в 2023г.
- ☐ 2й курс - поступившие в 2022г.
- ☐ 3й курс - поступившие в 2021г.
- ☐ 4й курс - поступившие в 2020г.
- ☐ 5й курс - поступившие в 2019г.

Для расчёта года выпуска к дате поступления необходимо прибавить пять лет обучения.



# Результат решения задачи

```
name_list = ['Аня', 'Дима', 'Света', 'Игорь', 'Андрей', 'Юля', 'Коля']  
s_names_list = ['Иванов(а)', 'Смирнов(а)', 'Кузнецов(а)', 'Соколов(а)', 'Петров(а)', 'Карпов(а)']
```

```
Введите количество студентов: 20  
0-['Юля', 'Карпов(а)', 2023, '1й курс']  
1-['Аня', 'Смирнов(а)', 2017, 'Выпустился в 2022 году']  
2-['Аня', 'Соколов(а)', 2019, '5й курс']  
3-['Дима', 'Соколов(а)', 2010, 'Выпустился в 2015 году']  
4-['Юля', 'Карпов(а)', 2011, 'Выпустился в 2016 году']  
5-['Коля', 'Иванов(а)', 2013, 'Выпустился в 2018 году']  
6-['Игорь', 'Смирнов(а)', 2016, 'Выпустился в 2021 году']  
7-['Игорь', 'Петров(а)', 2011, 'Выпустился в 2016 году']  
8-['Аня', 'Кузнецов(а)', 2021, '3й курс']  
9-['Юля', 'Петров(а)', 2018, 'Выпустился в 2023 году']  
10-['Света', 'Смирнов(а)', 2010, 'Выпустился в 2015 году']  
11-['Юля', 'Смирнов(а)', 2022, '2й курс']  
12-['Андрей', 'Карпов(а)', 2020, '4й курс']  
13-['Аня', 'Смирнов(а)', 2019, '5й курс']  
14-['Юля', 'Соколов(а)', 2023, '1й курс']  
15-['Игорь', 'Иванов(а)', 2023, '1й курс']  
16-['Света', 'Соколов(а)', 2016, 'Выпустился в 2021 году']  
17-['Юля', 'Иванов(а)', 2010, 'Выпустился в 2015 году']  
18-['Света', 'Соколов(а)', 2015, 'Выпустился в 2020 году']  
19-['Андрей', 'Соколов(а)', 2016, 'Выпустился в 2021 году']
```

Создать многомерный список из семи строк.

- ❑ 1 строка - заданное количество случайных, целых чисел в заданном диапазоне.
- ❑ 2-5 строки - разбить сгенерированные числа на отрицательные и положительные, а затем на чётные и нечетные.
- ❑ 6 строка - повторяющиеся значения.
- ❑ 7 строка - повторяющиеся по модулю значения.



# Результат решения задачи

Введите количество значений для генерации: 20

Введите минимальное число для генерации: -15

Введите максимальное число для генерации: 15

[-15, 2, -4, -8, 13, 0, -4, -6, 4, 2, -1, 2, 8, 9, 9, -8, -8, 12, -15, 8]

[-4, -8, 0, -4, -6, -8, -8]

[-15, -1, -15]

[2, 4, 2, 2, 8, 12, 8]

[13, 9, 9]

[-15, 2, -4, -8, 8, 9]

[-15, 2, -4, -8, 0, 4, 8, 9]



# Проверка решений