

6 WORKSHOP

PYTHON
PROGRAMMING
LANGUAGE

ANALYSIS OF TASKS FROM HOMEWORK

OBJECTIVES OF THE CONFERENCE

1. Напишите программу вычисления арифметического выражения заданного строкой. Используйте операции $+$, $-$, $/$, $*$ приоритет операций стандартный.
* Добавьте скобки, приоритет операций меняется.

```
in
>> 2 - 2 + 7 * 2

out
>> 14
```

```
in
>> 2 - ( 2 + 7 ) * 2

out
>> -16
```

```
in
>> 101 / 2 - ( 12 + 8 ) * 3

out
>> -9.5
```




HOMEWORK

1. Представлен список чисел. Необходимо вывести элементы исходного списка, значения которых больше предыдущего элемента. Use comprehension.

```
in
>> 10

out
>> [28, 20, 10, 5, 1, 24, 7, 15, 23, 25]
>> [24, 15, 23, 25]
```

```
in
>> 9

out
>> [15, 16, 2, 3, 1, 7, 5, 4, 10]
>> [16, 3, 7, 10]
```


2. Для чисел в пределах от 20 до N найти числа, кратные 20 или 21. Use comprehension.

```
in  
>> 100  
  
out  
>> [20, 21, 40, 42, 60, 63, 80, 84, 100]
```

```
in  
>> 424  
  
out  
>> [20, 21, 40, 42, 60, 63, 80, 84, 100,  
    105, 120, 126, 140, 147, 160, 168, 180,  
    189, 200, 210, 220, 231, 240, 252, 260,  
    273, 280, 294, 300, 315, 320, 336, 340,  
    357, 360, 378, 380, 399, 400, 420]
```


3. Написать функцию, аргументы – имена сотрудников, возвращает словарь, ключи – первые буквы имён, значения – списки, содержащие имена, начинающиеся с соответствующей буквы.

```
in
>> "Иван", "Мария", "Петр", "Илья",
    "Марина", "Петр", "Алина", "Бибочка"

out
>> {'А': ['Алина'], 'Б': ['Бибочка'],
    'И': ['Иван', 'Илья'], 'М': ['Марина', 'Мария'],
    'П': ['Петр', 'Петр']}
```


4. * Функция принимает в качестве аргументов строки в формате «Имя Фамилия», возвращает словарь, ключи – первые буквы фамилий, значения – словари, реализованные по схеме предыдущего задания.

```
in
>> "Иван Сергеев", "Инна Серова", "Петр Алексеев",
    "Илья Иванов", "Анна Савельева", "Юнона Ветрякова",
    "Борис Аркадьев", "Антон Серов", "Павел Анисимов"

out
>> {'С': {'А': ['Анна Савельева', 'Антон Серов'], 'И': ['Иван Сергеев', 'Инна Серова']},
     'А': {'Б': ['Борис Аркадьев'], 'П': ['Павел Анисимов', 'Петр Алексеев']},
     'И': {'И': ['Илья Иванов']}, 'В': {'Ю': ['Юнона Ветрякова']}}
```

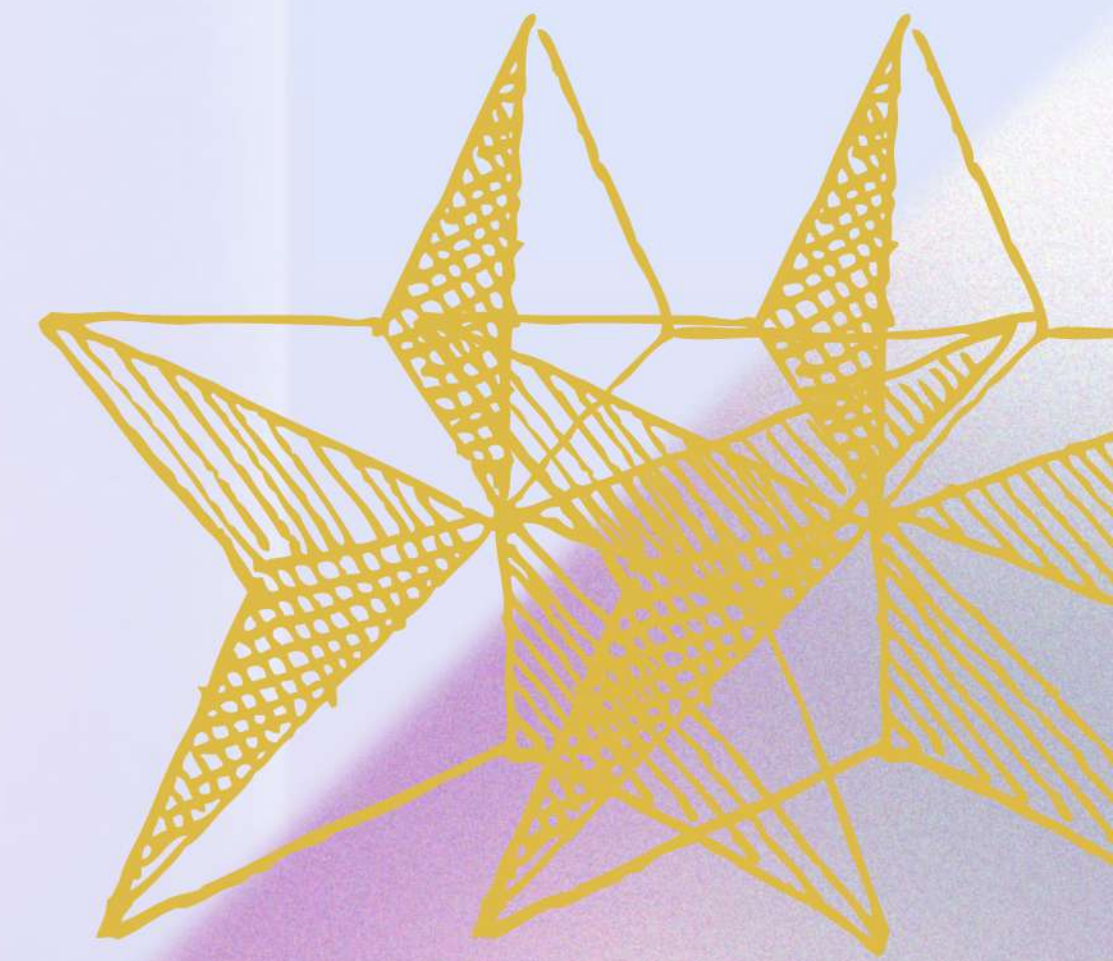


5. ** Реализовать функцию, возвращающую n шуток, сформированных из трех случайных слов, взятых из трёх списков (по одному из каждого).



```
in  
>> 10, True
```

```
out  
>> ['дом ночью мягкий', 'огонь завтра зеленый',  
     'лес вчера яркий', 'автомобиль сегодня веселый',  
     'город позавчера утопичный']
```



5. ** Реализовать функцию, возвращающую n шуток.



```
in  
>> 10, False
```

```
out
```

```
>> ['автомобиль ночью мягкий', 'огонь вчера веселый',  
    'автомобиль позавчера веселый', 'город вчера утопичный',  
    'лес сегодня зеленый', 'дом вчера яркий',  
    'автомобиль вчера зеленый', 'огонь позавчера яркий',  
    'огонь где-то утопичный', 'автомобиль где-то мягкий']
```

