

Задание 1.

Исходные данные:

Реализовать скрипт, в котором должна быть предусмотрена функция расчёта заработной платы сотрудника. Используйте в нём формулу: (выработка в часах*ставка в час) + премия. Во время выполнения расчёта для конкретных значений необходимо запускать скрипт с параметрами.

1. Решение:

```
a = 40
b = 1
c = 10
def my_func(a, b, c):
    a = int(input("Введите количество отработанных часов: "))
    b = int(input("Введите значение ставки: "))
    c = int(input("Введите значение премии: "))
    d = (a * b) + c
    return d
a = 40
b = 1
c = 10
def my_func(a, b, c):
    a = int(input("Введите количество отработанных часов: "))
    b = int(input("Введите значение ставки: "))
    c = int(input("Введите значение премии: "))
    d = (a * b) + c
    return d

print(my_func(a, b, c))
print(my_func(a, b, c))
Введите количество отработанных часов: 40
    Введите значение ставки: 2
Введите значение премии: 10
90
Введите количество отработанных часов: 80
    Введите значение ставки: 2
Введите значение премии: 10
170
```

Задание 2.

Исходные данные:

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

Подсказка: элементы, удовлетворяющие условию, оформить в виде списка. Для его формирования используйте генератор.

Пример исходного списка: [300, 2, 12, 44, 1, 1, 4, 10, 7, 1, 78, 123, 55].

Результат: [12, 44, 4, 10, 78, 123].

Решение:

Python 3.8.5 (default, Jan 27 2021, 15:41:15)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> my_list = [300, 2, 12, 44, 1, 1, 4, 10, 7, 1, 78, 123, 55]
```

```
>>> my_list.reverse()
```

```
>>> my_list_1 = [x for x in my_list if x > 2 and x < 150]
```

```
>>> my_list_1.reverse()
```

```
>>> my_list_1
```

```
[12, 44, 4, 10, 7, 78, 123, 55]
```

```
>>> del my_list_1[4]
```

```
>>> my_list_1
```

```
[12, 44, 4, 10, 78, 123, 55]
```

```
>>> del my_list_1[6]
```

```
>>> print(my_list_1)
```

```
[12, 44, 4, 10, 78, 123]
```

```
>>>
```

Задание 3.

Исходные данные:

Для чисел в пределах от 20 до 240 найти числа, кратные 20 или 21. Решите задание в одну строку.

Подсказка: используйте функцию `range()` и генератор.

Решение:

Python 3.8.5 (default, Jan 27 2021, 15:41:15)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> my_list = [20, 260]
```

```
>>> my_list_1 = [x for x in range(20, 260, 20)]
```

```
>>> print(my_list_1)
```

```
[20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240]
```

```
>>>
```

```
>>> my_list_2 = [x for x in range(20, 260, 21)]
```

```
>>> print(my_list_2)
```

```
[20, 41, 62, 83, 104, 125, 146, 167, 188, 209, 230, 251]
```

```
>>>
```

Задание 4.

Исходные данные:

Представлен список чисел. Определите элементы списка, не имеющие повторений. Сформируйте итоговый массив чисел, соответствующих требованию. Элементы выведите в порядке их следования в исходном списке. Для выполнения задания обязательно используйте генератор.

Пример исходного списка: [2, 2, 2, 7, 23, 1, 44, 44, 3, 2, 10, 7, 4, 11].

Результат: [23, 1, 3, 10, 4, 11].

Решение:

Python 3.8.5 (default, Jan 27 2021, 15:41:15)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> my_list = [2, 2, 2, 7, 23, 1, 44, 44, 3, 2, 10, 7, 4, 11]
```

```
>>> my_list_1 = [x for x in my_list if x < 2 or x > 2]
```

```
>>> my_list_2 = [x for x in my_list_1 if x < 7 or x > 7]
```

```
>>> my_list_3 = [x for x in my_list_2 if x < 40]
```

```
>>> print(my_list_3)
```

```
[23, 1, 3, 10, 4, 11]
```

```
>>>
```

Задание 5.

Исходные данные:

Реализовать формирование списка, используя функцию `range()` и возможности генератора. В список должны войти чётные числа от 100 до 1000 (включая границы). Нужно получить результат вычисления произведения всех элементов списка.

Подсказка: использовать функцию `reduce()`.

Решение:

Python 3.8.5 (default, Jan 27 2021, 15:41:15)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> my_list = [100, 1010]
```

```
>>> my_list_1 = [x for x in range(100, 1010, 10)]
```

```
>>> import functools
```

```
>>> from functools import reduce
```

```
>>> sn = reduce(lambda x,y: x * y, my_list_1)
```

```
>>> print(sn)
```

```
25718203109552511210785724993459738891841922471445552653382099838849647264448279
21322240519625124511856638500904630284343341744128000000000000000000000000000000
```

[illegible]

Задание 6.

Реализовать два небольших скрипта:

Подсказка: используйте функцию `count()` и `cycle()` модуля `itertools`. Обратите внимание, что создаваемый цикл не должен быть бесконечным. Предусмотрите условие его завершения.

Решение:

```
from itertools import count
```

```
print(i, end = ' ')
```

```
print()
```

```
break
```

```
import itertools
```

```
from itertools import cycle
```

```
x = list(range(5))
```

```
for i, j in enumerate(cycle(x)):
```

```
    print(j, end = ' ')
```

```
    if i > 13:
```

```
        print()
```

```
        break
```