

14.3 Функции: Задание 13

Напишите функцию, принимающей аргументы переменной длины и выводящей их значение

❖ Формат входных данных:

Произвольное количество аргументов

```
80 False [] () {key:value} {100} 'Superman'
```

❖ Формат выходных данных:

Одно значение тип `None`

Пример:

```
Введите аргументы через пробел: 80 False [] () {key:value} {100} 'Superman'
80
False
[]
()
{key:value}
{100}
'Superman'
Process finished with exit code 0
```

```
Введите аргументы через пробел:
Ошибка: Введите что-нибудь...
Process finished with exit code 0
```

14.3 Функции: Задание 14

Напишите функцию, которая принимает два целых числа и вычисляет их сумму и разность. Кроме того, функция должна возвращать как сложение, так и вычитание в одном и том же обратном вызове.

❖ Формат входных данных:

Два аргумента типа `int`

```
10 99
-45 20
```

❖ Формат выходных данных:

Два значения тип `int`

Пример:

```
Введите первое целое число: 10
Введите второе целое число: 99

summa = 109
difference = -89

Process finished with exit code 0
```

```
Введите первое целое число: -45
Введите второе целое число: 20

summa = -25
difference = -65

Process finished with exit code 0
```

14.3 Функции: Задание 15

Напишите функцию `show_employee()`, используя следующие условия:

- Она должна принимать имя и зарплату сотрудника и отображать их.
 - Если зарплата отсутствует в вызове функции, присвойте зарплате значение по умолчанию `50000`
- ❖ Формат входных данных:

Два аргумента типа `str, int`

```
Алёна 43000
Михаил
```

- ❖ Формат выходных данных:

Одно значение тип `None`

Пример:

```
Введите имя (обязательное поле): Алёна
Укажите зарплату (необязательное поле): 43000

Имя: Алёна
Зарплата: 43,000₽

Process finished with exit code 0
```

```
Введите имя (обязательное поле): Михаил
Укажите зарплату (необязательное поле):

Имя: Михаил
Зарплата: 50,000₽

Process finished with exit code 0
```

14.3 Функции: Задание 16

Напишите функцию, которая вычисляет сумму двух чисел и выводит результат, затем дайте функции другое имя и вызовите ее с новым именем.

❖ Формат входных данных:

Два аргумента типа `int`

```
34 10
```

❖ Формат выходных данных:

Одно значение тип `None`

Пример:

```
Введите первое число: 34
Введите второе число: 10

Результат первой функции: 44
Результат функции с новым именем: 44
Process finished with exit code 0
```

14.3 Функции: Задание 17

Создайте внешнюю и внутреннюю функции для вычисления сложения следующим образом:

- Внешняя функция будет принимать два аргумента, `a` и `b`.
- Внутренняя функция внутри внешней функции, которая вычисляет сложение `a` и `b` и возвращает результат.
- Наконец, внешняя функция добавит 10 к результату внутренней функции и вернет его.

❖ Формат входных данных:

Два аргумента типа `int`

```
90 50
-7 0
```

❖ Формат выходных данных:

Одно значение тип `int`

Пример:

```
Введите первое число: 90
Введите второе число: 50
150
Process finished with exit code 0
```

```
Введите первое число: -7
Введите второе число: 0
3
Process finished with exit code 0
```

14.3 Функции: Задание 18

Напишите функцию, подсчитывающую количество знаков текста без учёта пробелов в предложении, которое подаётся на вход в программу.

Добавьте и выведите строку документации: "Функция считает количество символов во входящей строке без учета пробелов"

❖ Формат входных данных:

Один аргумент типа `str`

```
В тот день с утра шел все тот же ледяной дождь
```

❖ Формат выходных данных:

Одно значение тип `int`

Пример:

```
Введите предложение: В тот день с утра шел все тот же ледяной дождь
Количество символов в предложении: 36

Функция считает количество символов во входящей строке без учета пробелов

:param sentence: исходный текст (предложение)
:return: возвращает количество символов без учета пробелов

Process finished with exit code 0
```

14.3 Функции: Задание 19

Отсортируйте словарь `data` по убыванию уязвимой версии, выведите результат, как на скриншоте

Примечание 1: Используйте `lambda` функцию

Примечание 2: Решение должно быть выполнено в одну строку

❖ Формат входных данных:

```
data = {'192.168.1.2': ('CVE-2016-8743', '2.2.32'),
        '192.168.1.12': ('CVE-2018-1283', '2.4.0'),
        '192.168.1.200': ('CVE-2022-23943', '2.4.52'),
        '192.168.1.5': ('CVE-2016-5387', '2.4.23')}
}
```

Пример:

```
'192.168.1.200', ('CVE-2022-23943', '2.4.52')
'192.168.1.5', ('CVE-2016-5387', '2.4.23')
'192.168.1.12', ('CVE-2018-1283', '2.4.0')
'192.168.1.2', ('CVE-2016-8743', '2.2.32')
Process finished with exit code 0
```

14.3 Функции: Задание 20

Напишите функцию:

- принимающую один аргумент, по умолчанию равный строке `"python"`
- в теле которой, присутствует конструкция, запрашивающая неограниченное количество слов
- после ввода пустой строки (нажатие Enter без ввода) распечатывается список слов
- если введённая строка уже есть в списке, завершить ввод и вывести сообщение `"Строка s уже присутствует в списке на позиции n"`

❖ Формат входных данных:

Один аргумент типа `str`

```
Orange kiwi django flask python
wolf tiger lion fox
```

❖ Формат выходных данных:

Одно значение тип `None`

Пример:

```
Укажите аргумент (необязательное поле):  
Введите любое слово:  
['python']
```

```
Process finished with exit code 0
```

```
Укажите аргумент (необязательное поле): Orange  
Введите любое слово: kiwi  
Введите любое слово: django  
Введите любое слово: flask  
Введите любое слово: python  
Введите любое слово: django  
Строка "django" уже присутствует в списке на позиции "2"
```

```
Process finished with exit code 0
```

```
Укажите аргумент (необязательное поле):  
Введите любое слово: wolf  
Введите любое слово: tiger  
Введите любое слово: lion  
Введите любое слово: fox  
Введите любое слово: python  
Строка "python" уже присутствует в списке на позиции "0"
```

```
Process finished with exit code 0
```