

## Задания #23

**#1.** Вводятся данные в формате ключ=значение в одну строчку через пробел. Значениями здесь являются целые числа (см. пример ниже). Необходимо на их основе создать словарь d с помощью функции dict() и вывести его на экран командой:

```
print(*sorted(d.items()))
```

---

### Sample Input:

```
one=1 two=2 three=3
```

---

### Sample Output:

```
('one', 1) ('three', 3) ('two', 2)
```

**#2.** На вход программы поступают данные в виде набора строк в формате:

```
ключ1=значение1  
ключ2=значение2  
...  
ключN=значениеN
```

Ключами здесь выступают целые числа (см. пример ниже). Необходимо их преобразовать в словарь d (без использования функции dict()) и вывести его на экран командой:

```
print(*sorted(d.items()))
```

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

---

### Sample Input:

```
5=отлично  
4=хорошо  
3=удовлетворительно
```

---

### Sample Output:

```
(3, 'удовлетворительно') (4, 'хорошо') (5, 'отлично')  
  
import sys  
  
# считывание списка из входного потока  
lst_in = list(map(str.strip, sys.stdin.readlines()))
```

**#3.** Вводятся данные в формате ключ=значение в одну строчку через пробел. Необходимо на их основе создать словарь, затем проверить, существуют ли в нем ключи со значениями: 'house', 'True' и '5' (все ключи - строки). Если все они существуют, то вывести на экран ДА, иначе - НЕТ.

---

### Sample Input:

```
вологда=город house=дом True=1 5=отлично 9=божественно
```

---

**Sample Output:**

ДА

**#4.** Вводятся данные в формате ключ=значение в одну строчку через пробел. Необходимо на их основе создать словарь d, затем удалить из этого словаря ключи 'False' и '3', если они существуют. Ключами и значениями словаря являются строки. Вывести полученный словарь на экран командой:

```
print(*sorted(d.items()))
```

---

**Sample Input:**

лена=имя дон=река москва=город False=ложь 3=удовлетворительно  
True=истина

---

**Sample Output:**

```
('True', 'истина') ('дон', 'река') ('лена', 'имя') ('москва',  
'город')
```

**#5.** Вводятся номера телефонов в одну строчку через пробел с разными кодами стран: +7, +6, +2, +4 и т.д. Необходимо составить словарь d, где ключи - это коды +7, +6, +2 и т.п., а значения - список номеров (следующих в том же порядке, что и во входной строке) с соответствующими кодами. Полученный словарь вывести командой:

```
print(*sorted(d.items()))
```

---

**Sample Input:**

+71234567890 +71234567854 +61234576890 +52134567890  
+21235777890 +21234567110 +71232267890

---

**Sample Output:**

```
('+2', ['+21235777890', '+21234567110']) ('+5',  
['+52134567890']) ('+6', ['+61234576890']) ('+7',  
['+71234567890', '+71234567854', '+71232267890'])
```

**#6.** Вводятся номера телефонов в формате:

номер\_1 имя\_1  
номер\_2 имя\_2  
...  
номер\_N имя\_N

Необходимо создать словарь d, где ключами будут имена, а значениями - список номеров телефонов для этого имени. Обратите внимание, что одному имени может принадлежать несколько разных номеров. Полученный словарь вывести командой:

```
print(*sorted(d.items()))
```

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

---

**Sample Input:**

```
+71234567890 Сергей
+71234567810 Сергей
+51234567890 Михаил
+72134567890 Николай
```

---

**Sample Output:**

```
('Михаил', ['+51234567890']) ('Николай', ['+72134567890'])
('Сергей', ['+71234567890', '+71234567810'])
```

**#7.** Пользователь вводит в цикле целые положительные числа, пока не введет число 0. Для каждого числа вычисляется квадратный корень (с точностью до сотых) и значение выводится на экран (в столбик). С помощью словаря выполните кэширование данных так, чтобы при повторном вводе того же самого числа результат не вычислялся, а бралось ранее вычисленное значение из словаря. При этом на экране должно выводиться:

значение из кэша: <число>

---

**Sample Input:**

```
1
2
3
3
2
4
0
```

---

**Sample Output:**

```
1.0
1.41
1.73
значение из кэша: 1.73
значение из кэша: 1.41
2.0
```

**#8.** Тестовый веб-сервер возвращает HTML-страницы по URL-адресам (строкам). На вход программы поступают различные URL-адреса. Если адрес пришел впервые, то на экране отобразить строку (без кавычек):

"HTML-страница для адреса <URL-адрес>"

Если адрес приходит повторно, то следует взять строку "HTML-страница для адреса <URL-адрес>" из словаря и вывести на экран сообщение (без кавычек):

"Взято из кэша: HTML-страница для адреса <URL-адрес>"

Сообщения выводить каждое с новой строки.

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

---

**Sample Input:**

```
ustanovka-i-zapusk-yazyka
ustanovka-i-poryadok-raboty-pycharm
peremennyye-operator-prisvaivaniya-tipy-dannykh
arifmeticheskiye-operatsii
ustanovka-i-poryadok-raboty-pycharm
```

---

**Sample Output:**

```
HTML-страница для адреса ustanovka-i-zapusk-yazyka
HTML-страница для адреса ustanovka-i-poryadok-raboty-pycharm
HTML-страница для адреса peremennyye-operator-prisvaivaniya-
tipy-dannykh
HTML-страница для адреса arifmeticheskiye-operatsii
Взято из кэша: HTML-страница для адреса ustanovka-i-poryadok-
raboty-pycharm
```

```
import sys

lst_in = list(map(str.strip, sys.stdin.readlines()))
```