

## Лабораторная работа № 4

### Python: списки, множества, словари (основы)

#### Содержание

1 Цель работы.....	2
2 Краткие теоретические сведения.....	2
3 Оборудование рабочего места.....	2
4 Задание.....	2
5 Содержание отчета.....	2
6 Порядок выполнения работы.....	3
6.1 Списки.....	3
6.2 Операции со списками.....	6
6.3 Методы списков.....	10
6.4 Понятия множества и словаря.....	11
6.5 Самостоятельная работа.....	12
6.6 Дополнительное задание.....	13

## **1 Цель работы**

Изучить типовые случаи применения списков и словарей, а также порядок работы с ними в языке программирования Python.

## **2 Краткие теоретические сведения**

Программирование — это процесс проектирования, написания и отладки программ. Что такое программа? Программа — это последовательность инструкций, предназначенная для исполнения устройством управления вычислительной машины [1]. Есть, также, официальные трактовки термина «Программа»:

Программа — данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определённого алгоритма. [2]

Программа — представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определённого результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения. [3]

## **3 Оборудование рабочего места**

Персональный компьютер под управлением ОС Linux или Windows с установленной средой программирования IDLE и языком программирования Python версии 3.4 и выше.

## **4 Задание**

- Ознакомиться с определениями «список», «словарь».
- Ознакомиться с возможными областями использования списков и словарей.
- Выполнить примеры, приведенные в разделе «Порядок выполнения работы».
- Решить примеры для самостоятельной работы.

## **5 Содержание отчета**

Отчет представляется в виде текстового документа, содержащего скриншоты окна редактора, содержащего код программы и командного окна с результатом выполнения программы для всех самостоятельных заданий.

## 6 Порядок выполнения работы

### 6.1 Списки

Списки — это важная структура данных языка программирования Python.

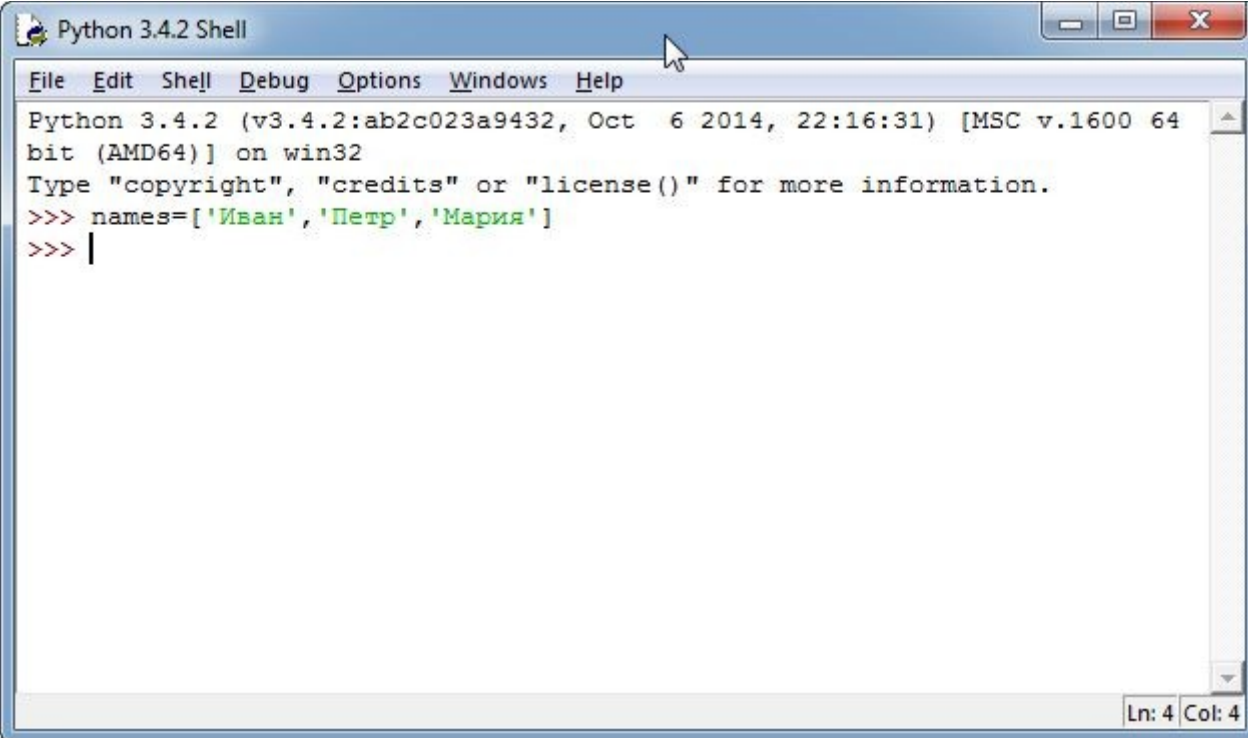
Список — это упорядоченный набор данных. В качестве элемента списка могут выступать данные разных типов: строка, целое число, число с плавающей точкой или даже другой список. В последнем случае принято говорить о «вложенных» списках.

Работа с элементами списка, во многом, напоминает работу с элементами строки: каждый элемент имеет свой индекс, можно выполнить цикл `for` для всех элементов списка.

В языке Python список обозначается квадратными скобками.

Рассмотрим пример создания и использования списка.

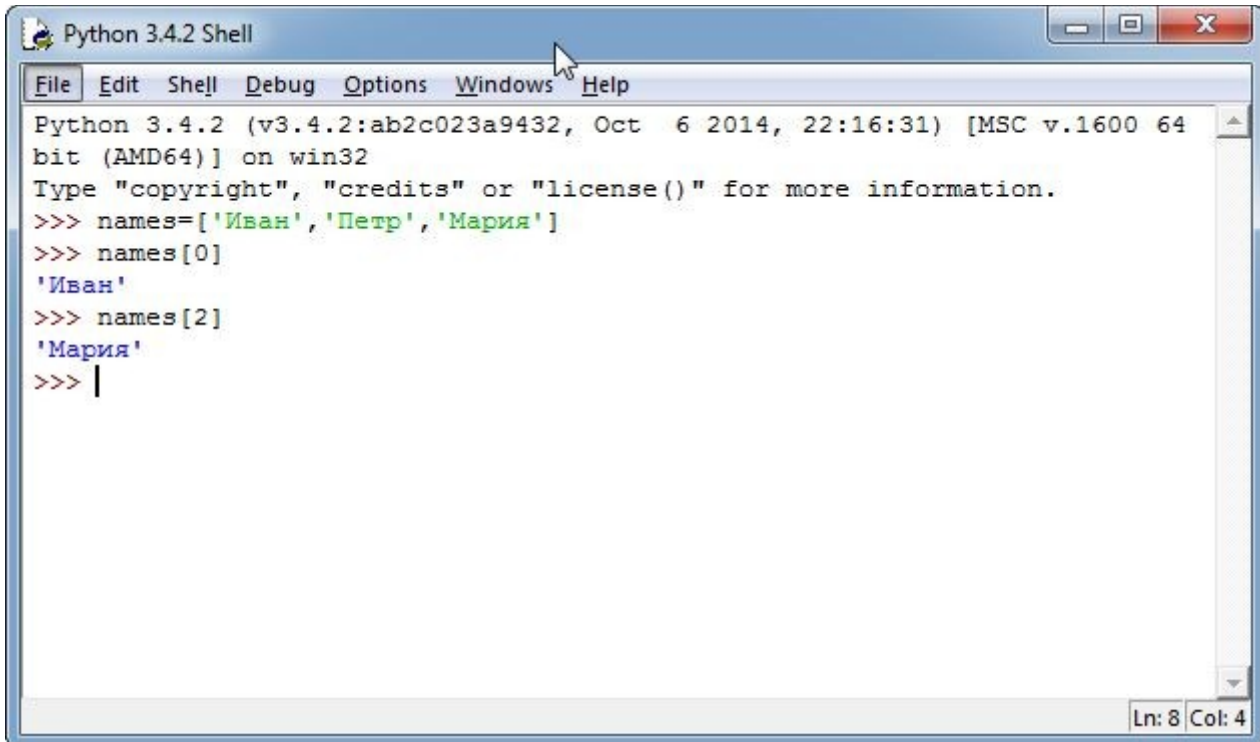
Создать список очень просто: необходимо перечислить его элементы через запятую в квадратных скобках:

A screenshot of a Python 3.4.2 Shell window. The window has a title bar with the text "Python 3.4.2 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Windows, and Help. The main area of the window is a text editor showing the following text:

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64  
bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> names=['Иван', 'Петр', 'Мария']  
>>> |
```

The text is displayed in a monospaced font. The first line of the prompt shows the Python version and build information. The second line prompts the user to enter "copyright", "credits", or "license()". The third line shows the user entering the list creation command. The fourth line shows the prompt character ">>>" followed by a vertical bar "|". The status bar at the bottom right of the window shows "Ln: 4 Col: 4".

После этого к элементам списка можно обращаться по их индексам:

A screenshot of a Python 3.4.2 Shell window. The title bar reads "Python 3.4.2 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following code:

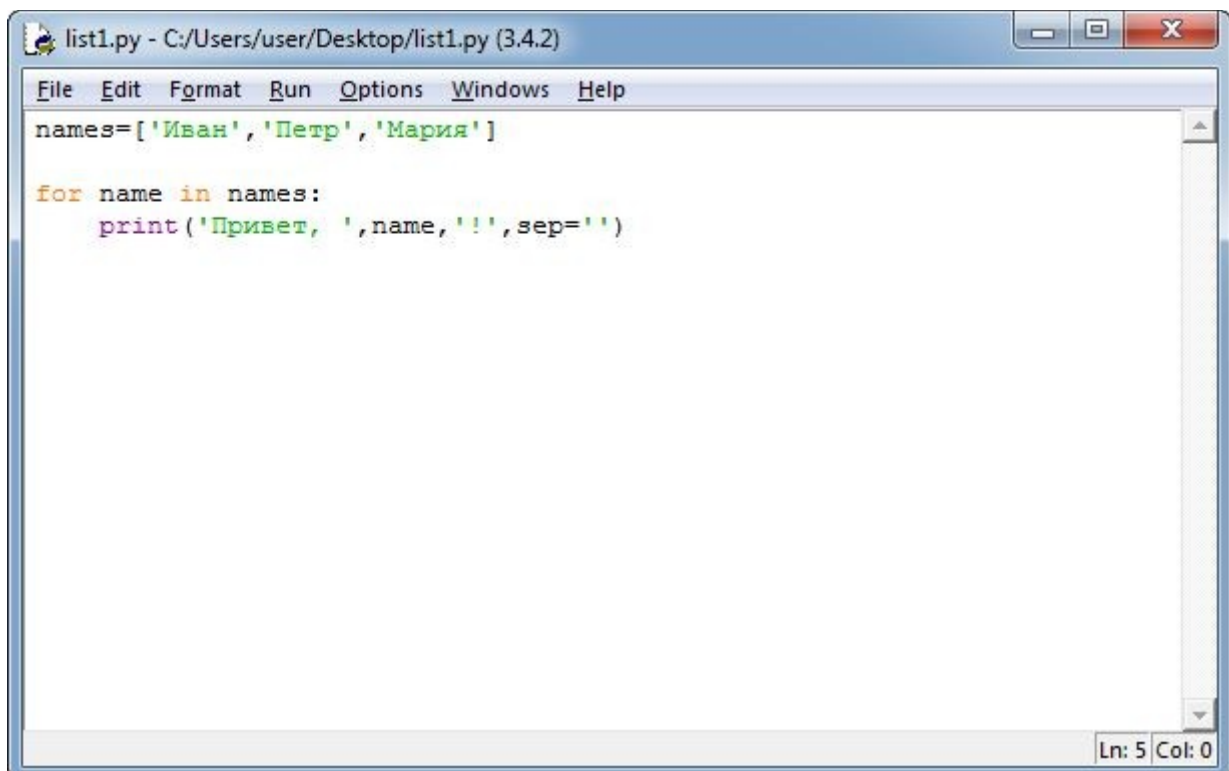
```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> names=['Иван', 'Петр', 'Мария']
>>> names[0]
'Иван'
>>> names[2]
'Мария'
>>> |
```

The status bar at the bottom right indicates "Ln: 8 Col: 4".

Все элементы списка можно обойти в цикле `for`. При этом в общем виде цикл будет выглядеть следующим образом:

```
for [переменная] in [список]:
    [действие1]
    [действие2]
```

Например:

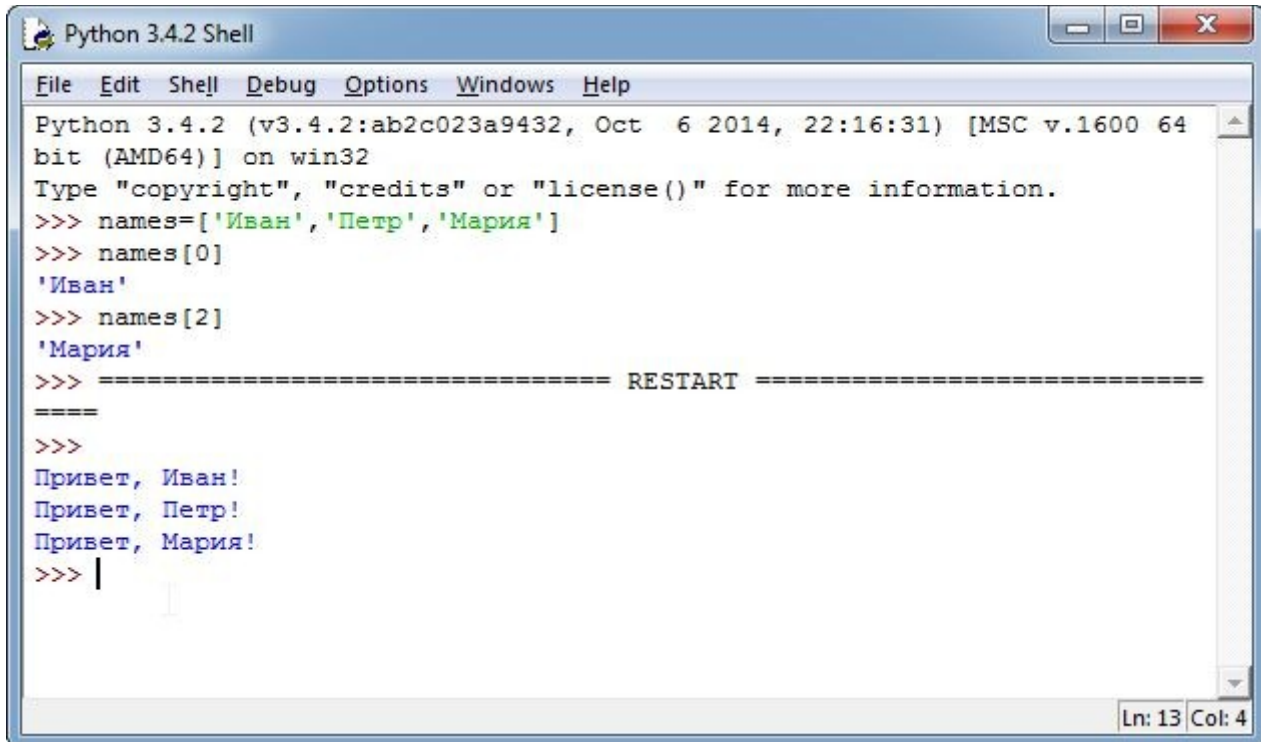
A screenshot of a Python script editor window titled "list1.py - C:/Users/user/Desktop/list1.py (3.4.2)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area shows the following code:

```
names=['Иван', 'Петр', 'Мария']

for name in names:
    print('Привет, ', name, '!', sep='')
```

The status bar at the bottom right indicates "Ln: 5 Col: 0".

Результат выполнения программы:

A screenshot of a Python 3.4.2 Shell window. The window has a title bar with the text "Python 3.4.2 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Windows, and Help. The main area of the window contains the following text:

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> names=['Иван', 'Петр', 'Мария']
>>> names[0]
'Иван'
>>> names[2]
'Мария'
>>> ===== RESTART =====
=====
>>>
Привет, Иван!
Привет, Петр!
Привет, Мария!
>>> |
```

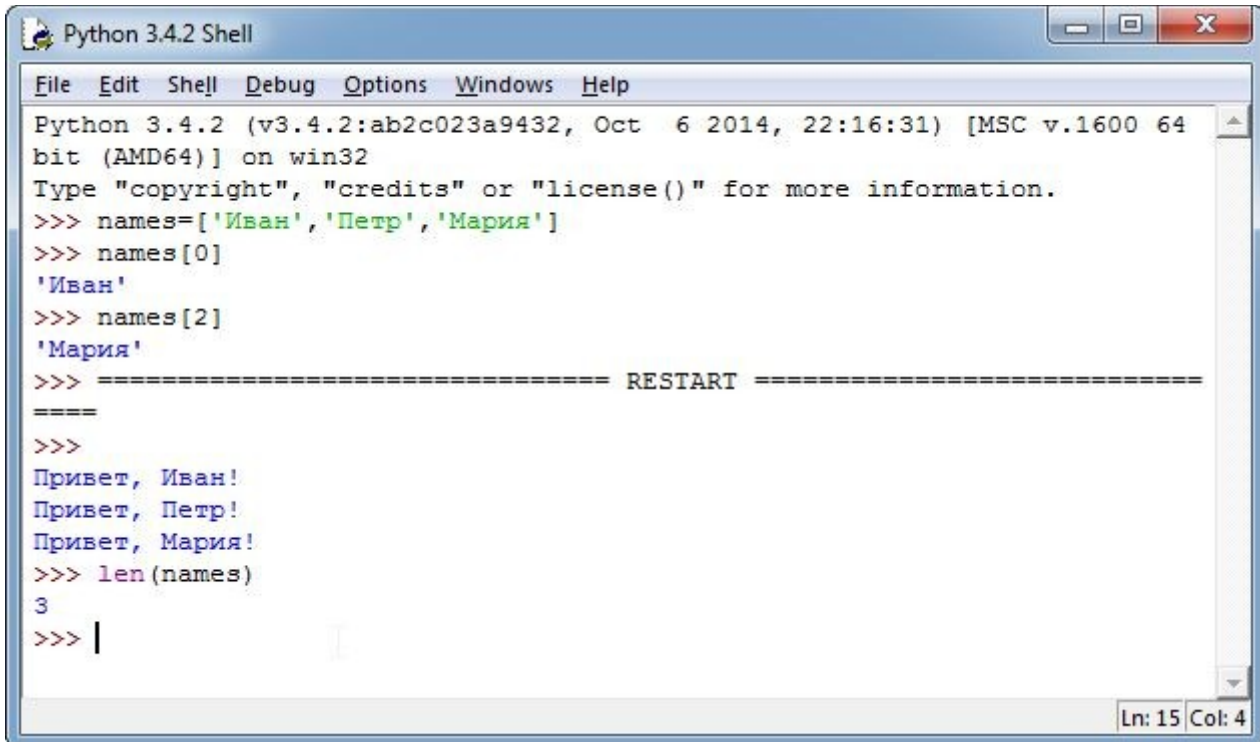
The text is displayed in a monospaced font. The names in the list and the output strings are in Cyrillic. The prompt characters are green. The status bar at the bottom right of the window shows "Ln: 13 Col: 4".

## 6.2 Операции со списками

Рассмотрим основные операции, которые можно выполнять со списками.

### Определение длины списка

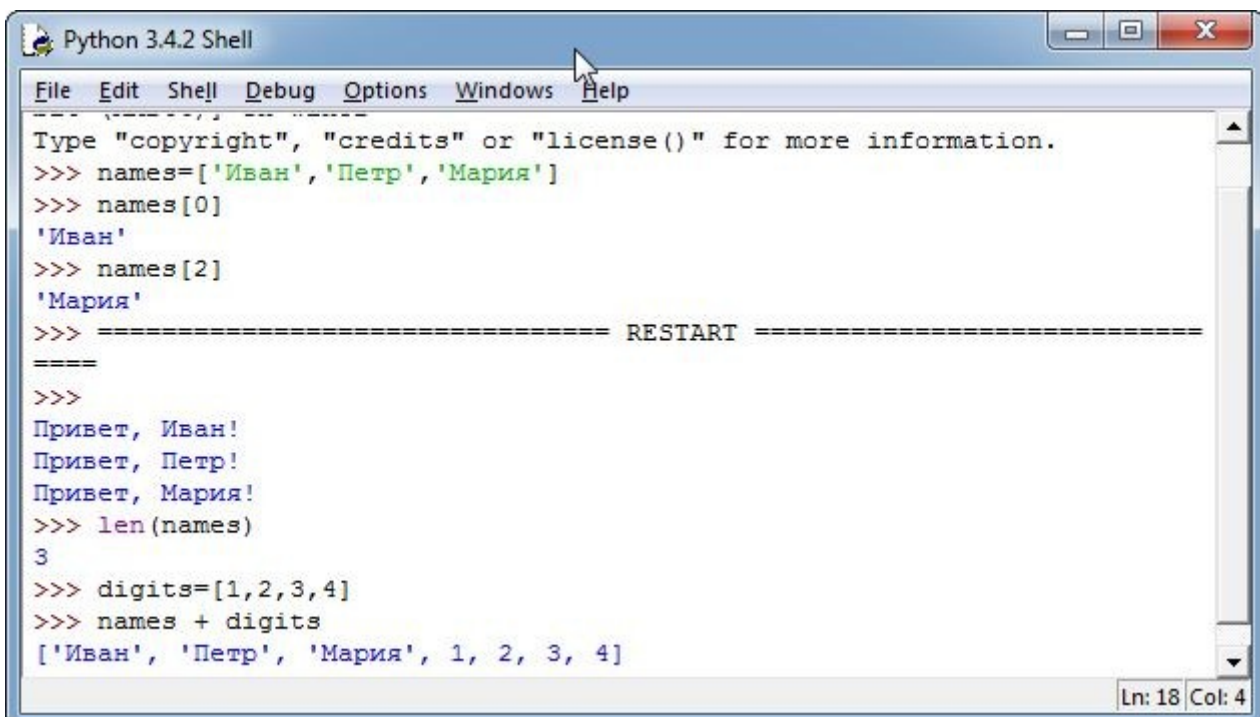
Для того, чтобы определить длину списка, используется та же функция, что и для определения длины строки — `len`. Например:



```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> names=['Иван', 'Петр', 'Мария']
>>> names[0]
'Иван'
>>> names[2]
'Мария'
>>> ===== RESTART =====
>>>
Привет, Иван!
Привет, Петр!
Привет, Мария!
>>> len(names)
3
>>> |
```

### Сложение списков

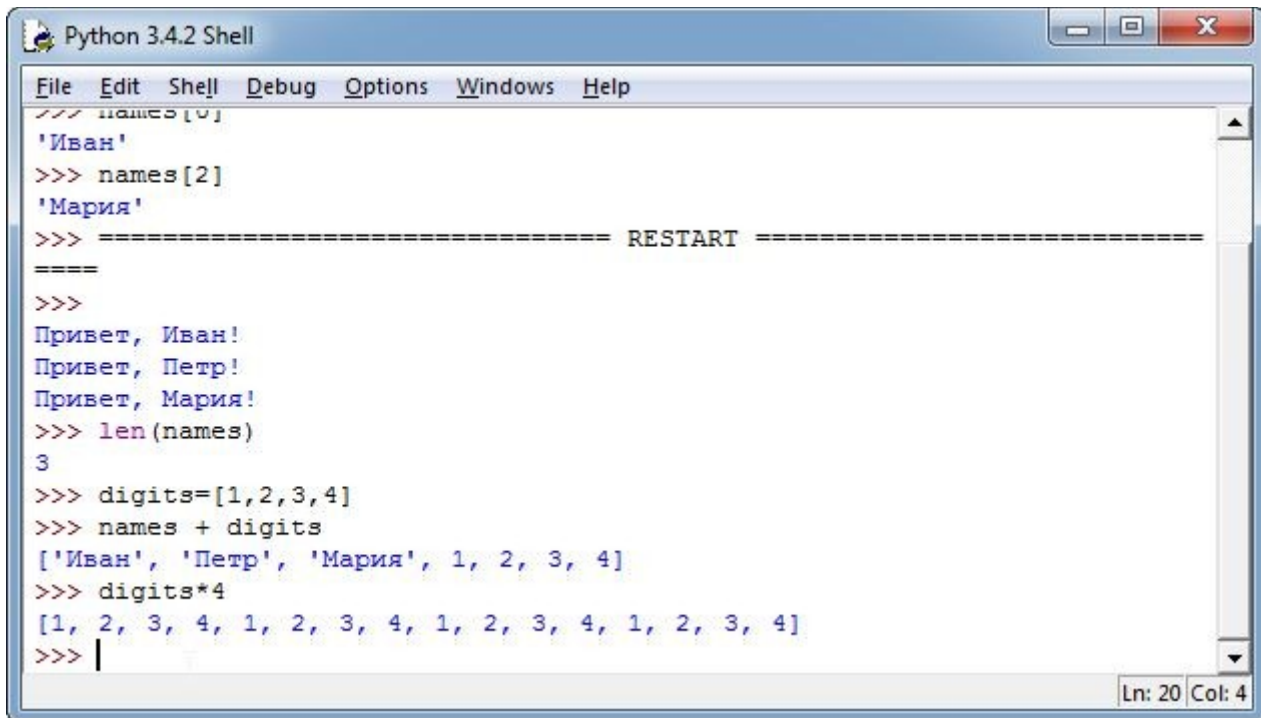
При сложении списков элементы второго списка располагаются за элементами первого списка, например:



```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Type "copyright", "credits" or "license()" for more information.
>>> names=['Иван', 'Петр', 'Мария']
>>> names[0]
'Иван'
>>> names[2]
'Мария'
>>> ===== RESTART =====
>>>
Привет, Иван!
Привет, Петр!
Привет, Мария!
>>> len(names)
3
>>> digits=[1,2,3,4]
>>> names + digits
['Иван', 'Петр', 'Мария', 1, 2, 3, 4]
```

## Умножение списка на целое число

Как и при умножении строки, приводит к повторению списка, например:

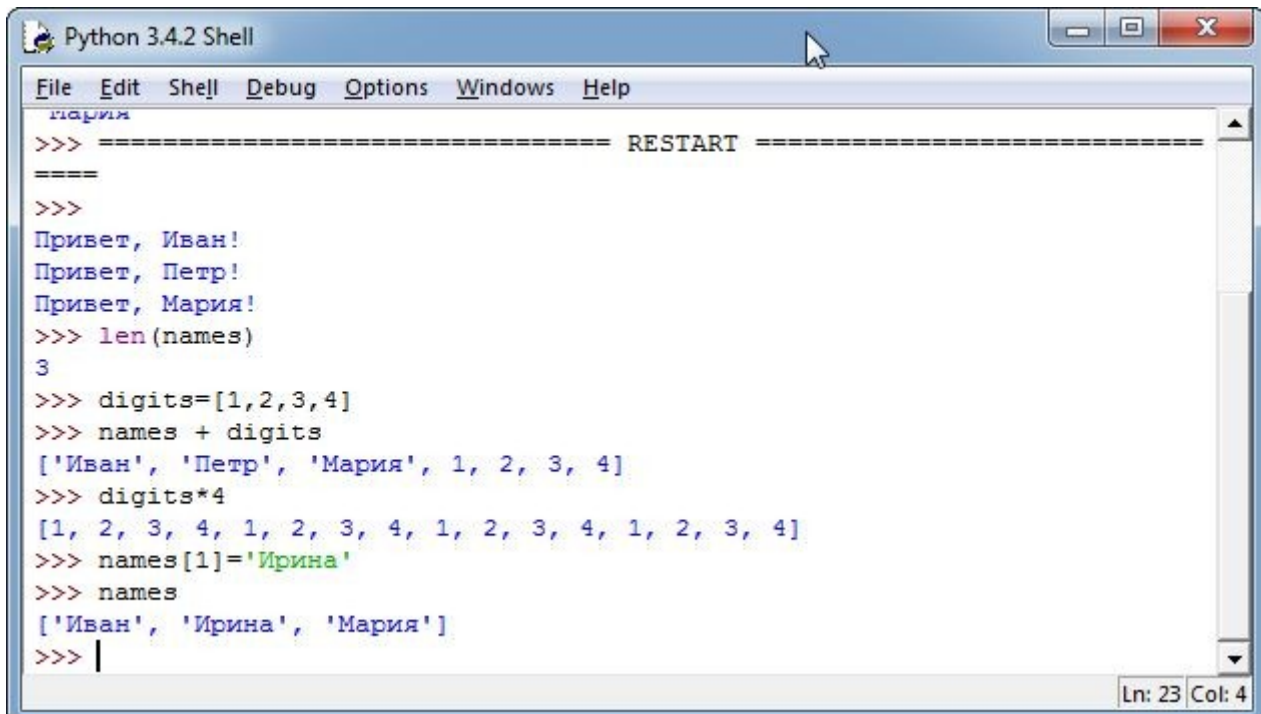


```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
>>> names = ['Иван', 'Петр', 'Мария']
>>> names[2]
'Мария'
>>> ===== RESTART =====
>>>
>>> Привет, Иван!
>>> Привет, Петр!
>>> Привет, Мария!
>>> len(names)
3
>>> digits = [1, 2, 3, 4]
>>> names + digits
['Иван', 'Петр', 'Мария', 1, 2, 3, 4]
>>> digits*4
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
>>> |
```

Ln: 20 Col: 4

## Изменение элемента списка

Для изменения элемента списка необходимо выполнить операцию присваивания с указанием индекса элемента, например:



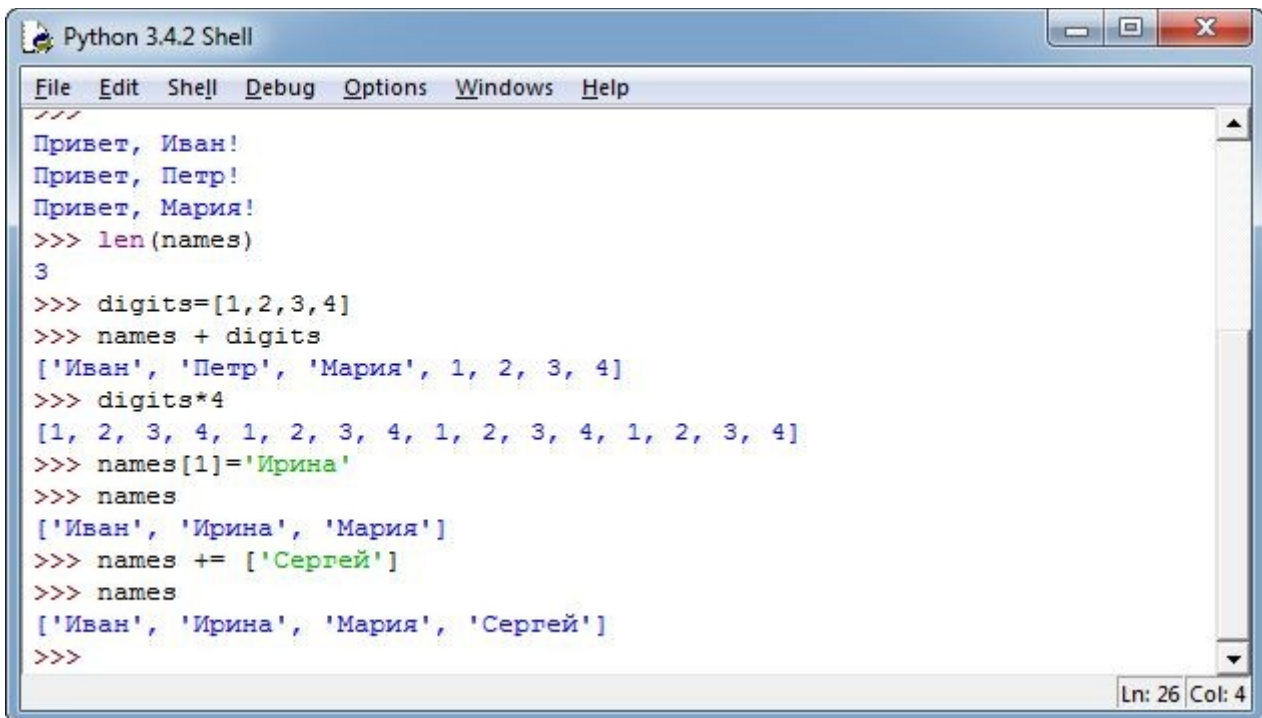
```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
>>> ===== RESTART =====
>>>
>>> Привет, Иван!
>>> Привет, Петр!
>>> Привет, Мария!
>>> len(names)
3
>>> digits = [1, 2, 3, 4]
>>> names + digits
['Иван', 'Петр', 'Мария', 1, 2, 3, 4]
>>> digits*4
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
>>> names[1] = 'Ирина'
>>> names
['Иван', 'Ирина', 'Мария']
>>> |
```

Ln: 23 Col: 4



### Добавление элемента к списку

Для добавления элемента к списку используют оператор `+=` и квадратные скобки. Например:

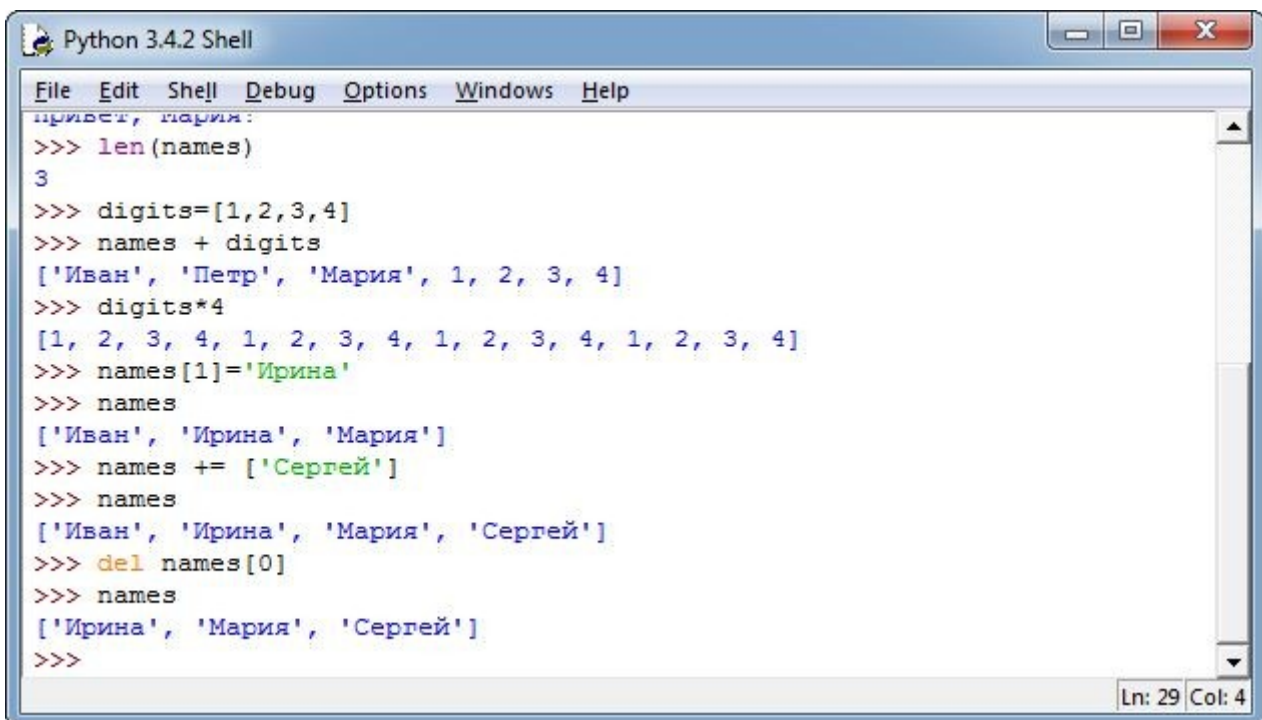


```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Привет, Иван!
Привет, Петр!
Привет, Мария!
>>> len(names)
3
>>> digits=[1,2,3,4]
>>> names + digits
['Иван', 'Петр', 'Мария', 1, 2, 3, 4]
>>> digits*4
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
>>> names[1]='Ирина'
>>> names
['Иван', 'Ирина', 'Мария']
>>> names += ['Сергей']
>>> names
['Иван', 'Ирина', 'Мария', 'Сергей']
>>>
```

Ln: 26 Col: 4

### Удаление элемента из списка

Для удаления элемента из списка используются функция `del` и индекс элемента. Например:



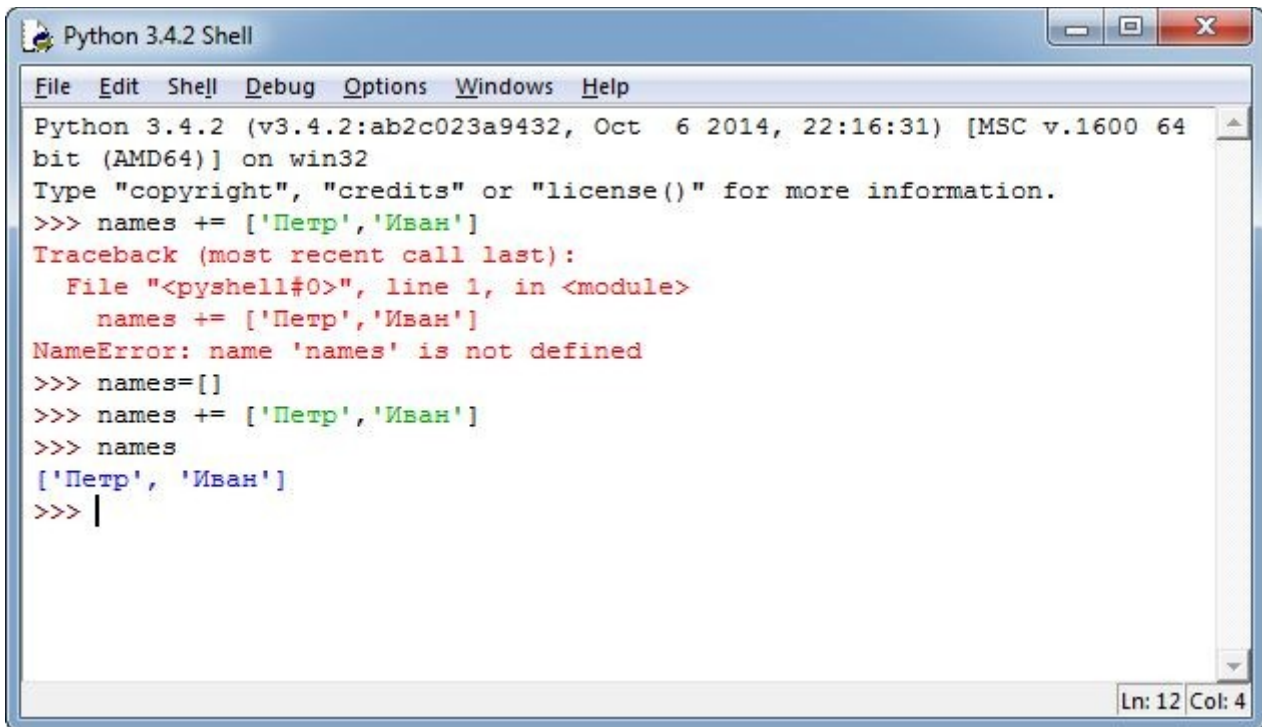
```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Привет, Мария!
>>> len(names)
3
>>> digits=[1,2,3,4]
>>> names + digits
['Иван', 'Петр', 'Мария', 1, 2, 3, 4]
>>> digits*4
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
>>> names[1]='Ирина'
>>> names
['Иван', 'Ирина', 'Мария']
>>> names += ['Сергей']
>>> names
['Иван', 'Ирина', 'Мария', 'Сергей']
>>> del names[0]
>>> names
['Ирина', 'Мария', 'Сергей']
>>>
```

Ln: 29 Col: 4



## Создание пустого списка

Перед добавлением элементов в список его необходимо создать. В противном случае возникнет ошибка. Для создания пустого списка также используются квадратные скобки. Например:



```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> names += ['Петр', 'Иван']
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    names += ['Петр', 'Иван']
NameError: name 'names' is not defined
>>> names=[]
>>> names += ['Петр', 'Иван']
>>> names
['Петр', 'Иван']
>>> |
```

## Поиск минимального и максимального значения

Для поиска минимального и максимального значения списка используются функции `min` и `max`. Для их применения список должен состоять из данных одного типа.

### **6.3 Методы списков**

Подобно строкам, списки имеют свои методы. Рассмотрим некоторые из них:

- `список.append('элемент')` - добавление элементов к списку,
- `список.insert(n, 'элемент')` - вставка элементов в список (сдвиг, а не замена существующих),
- `список.remove('элемент')` - удаляет первое вхождение элемента,
- `список.sort()` - сортировка списка (с сохранением),
- `список.reverse()` - развернуть (с сохранением).

#### ***6.4 Понятия множества и словаря***

Множество — это особый тип списков, значения в которых не повторяются, а элементы не имеют порядка.

Создание пустого множества:

```
n=set()
```

Создание множества:

```
множество={элемент1, элемент2, элемент3}
```

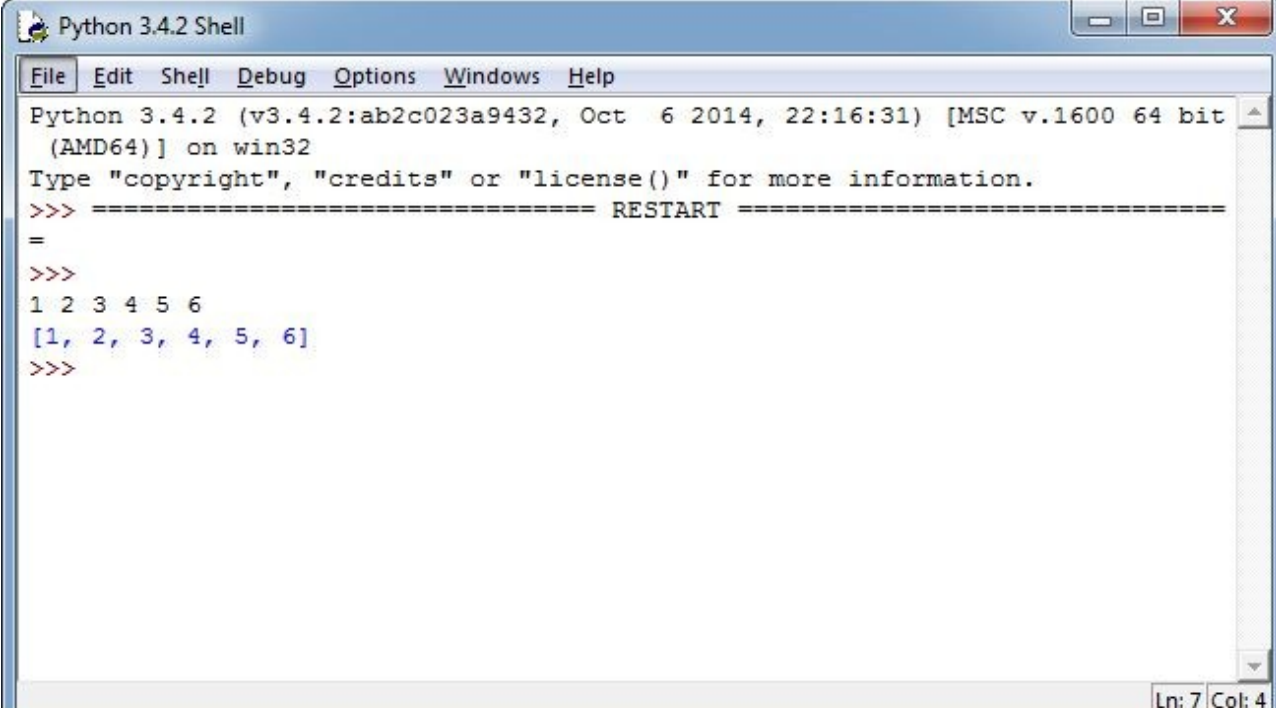
Некоторые методы множеств:

- `множество.add(element)` -добавить элемент,
- `множество.remove(element)` — удалить элемент, если нет элемента, возникает ошибка
- `множество.discard(element)` — удалить элемент, если нет элемента, ошибка не возникает,
- `множество.clear()` - очистить множество.

Словарь — это множество пар "ключ-значение", элементы которого не имеют порядка.

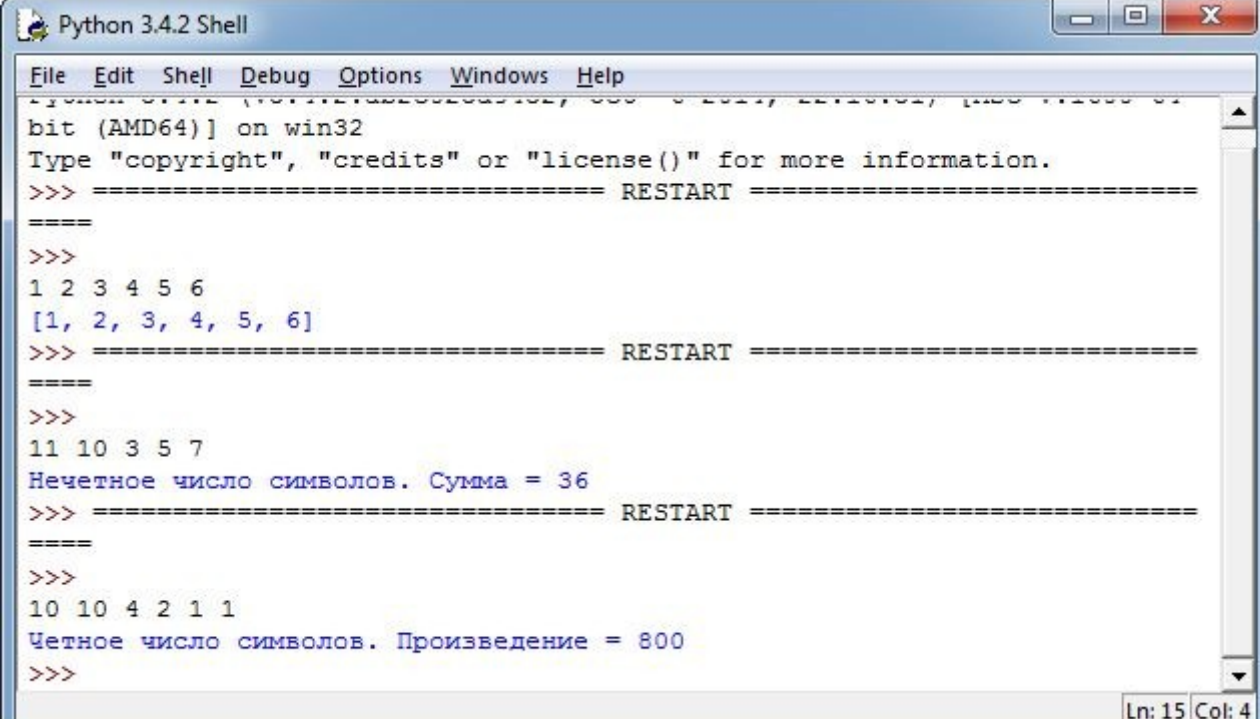
### 6.5 Самостоятельная работа

1. Напишите программу, на вход которой подается строка из целых чисел, разделенных пробелом. Необходимо вывести список, состоящий из этих чисел (не из строк!). Например:



```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
1 2 3 4 5 6
[1, 2, 3, 4, 5, 6]
>>>
```

2. Напишите программу, на вход которой подается строка из целых чисел, разделенных пробелом. Необходимо вывести произведение введенных чисел, если введено четное количество чисел, или сумму введенных чисел, если их количество нечетное. Например:



```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
1 2 3 4 5 6
[1, 2, 3, 4, 5, 6]
>>> ===== RESTART =====
>>>
11 10 3 5 7
Нечетное число символов. Сумма = 36
>>> ===== RESTART =====
>>>
10 10 4 2 1 1
Четное число символов. Произведение = 800
>>>
```

### 6.6 Дополнительное задание

Написать программу для решения следующей задачи:

Ввод состоит из 2-х строк.

Первая строка состоит из числа  $n$  ( $1 \leq n \leq 10^5$ ) и  $n$  натуральных чисел (не больше  $10^9$ ), разделенных пробелами в порядке возрастания.

Вторая строка из числа  $k$  ( $1 \leq k \leq 10^5$ ) и  $k$  натуральных чисел (не больше  $10^9$ ), разделенных пробелами.

Выполнить следующие операции:

- сформировать список  $N$ , состоящий из чисел, введенных в первой строке (после  $n$ ),
- сформировать список  $K$ , состоящий из чисел, введенных в первой строке (после  $k$ ),
- для каждого элемента списка  $K$  проверить его наличие в списке  $N$ , если он присутствует в списке, вывести позицию элемента в списке  $N$ , если отсутствует — вывести «-1».

Пример ввода 1:

```
5 1 5 8 12 13
5 8 1 23 1 11
```

Пример вывода 1:

```
3 1 -1 1 -1
```

Пример ввода 2:

```
5 1 2 3 4 5
10 10 9 8 7 6 5 4 3 2 1
```

Пример вывода 2:

```
-1 -1 -1 -1 -1 5 4 3 2 1
```

**Примечание:** при решении задачи можно использовать словари.