

ИНФОРМАТИКА

8

класс

ОРГАНИЗАЦИЯ ВВОДА И ВЫВОДА ДАННЫХ

НАЧАЛА ПРОГРАММИРОВАНИЯ
НА ЯЗЫКЕ PYTHON

КЛЮЧЕВЫЕ СЛОВА

- ◆ оператор вывода `print()`
- ◆ формат вывода
- ◆ оператор ввода `input()`

КАК УВИДЕТЬ РЕЗУЛЬТАТ?

Вы рассмотрели оператор присваивания, научились задавать значения переменных.



Вы можете записать программу преобразования данных.
Но результат этих преобразований виден не будет.

Для вывода данных из оперативной памяти на экран нужен специальный оператор!



ВЫВОД ДАННЫХ

Вывод данных из оперативной памяти на экран:

```
print(<выражение 1> ,< выражение 2> , ...,< выражение N>)
```

СПИСОК ВЫВОДА

Выражения - символьные, числовые, логические, в том числе переменные и константы

Пример:

```
s = 10
```

```
print ('s=' , s)
```

На экране:

```
s= 10
```

Информация в кавычках выводится на экран без изменений

ВАРИАНТЫ ОРГАНИЗАЦИИ ВЫВОДА

Вариант организации вывода	Оператор (функция) вывода	Результат
По умолчанию	<code>print(1, 20, 300)</code>	1 □ 20 □ 300
Убрать разделители-пробелы	<code>print(1, 20, 300, sep='')</code>	120300
Добавить разделитель-запятую	<code>print(1, 20, 300, sep=', □')</code>	1, □ 20, □ 300
Вывод каждого значения с новой строки	<code>print(1, 20, 300, sep='\n')</code>	1 20 30



Обратите внимание: оператор `print` вставляет между выводимыми значениями разделитель (сепаратор, от англ. *separator*). По умолчанию разделителем между выводимыми выражениями является один пробел. Его можно изменять, указывая новый разделитель после слова `sep`.

ФОРМАТ ВЫВОДА

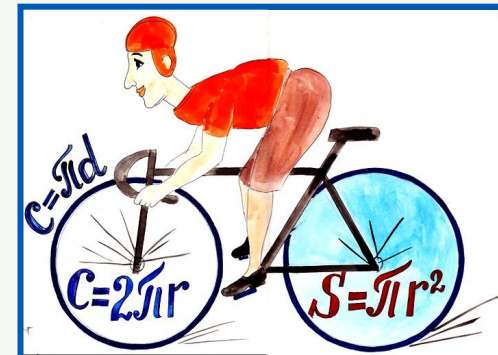
Формат вывода позволяет установить количество позиций на экране, занимаемых выводимой величиной.

d - вывод целого значения

f - вывод вещественного значения

Оператор вывода	Результат выполнения оператора
<pre>a = 4 print("a=", "{:5d}{:5d}".format(a, a*a))</pre>	<pre>a= 4 16</pre>
<pre>a =1/3 b =1/9 print("{:7.3f}{:7.3f}".format(a, b))</pre>	<pre> 0.333 0.111</pre>

ПЕРВАЯ ПРОГРАММА



```
r = 5.4
c = 2 * 3.14 * r
s = 3.14 * r * r
print ('c=', c)
print ('s=', s)
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py ====
c= 33.912000000000006
s= 91.562400000000003
>>> |
```

Ln: 7 Col: 4

ОТЛАДКА ПРОГРАММЫ

Синтаксическая ошибка

```
proba.py - C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py (3.9.0)
File Edit Format Run Options Window Help
r = 5.4
c = 2 * 3.14 * r
s = 3.14 * r * r
prent ('c=', c)
print ('s=', s)

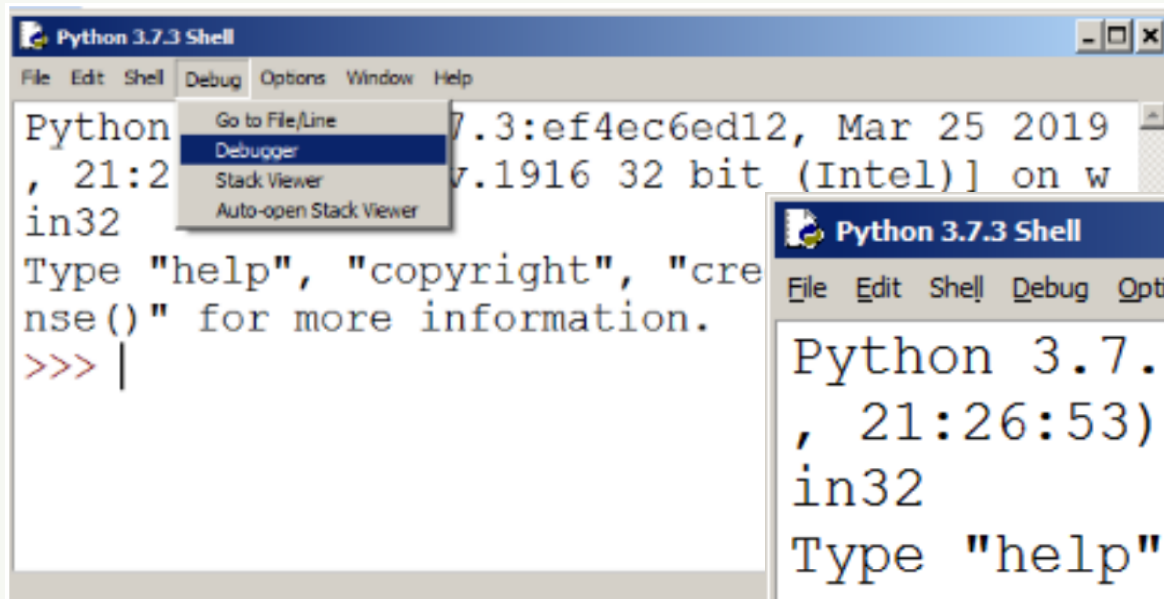
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
>>>
==== RESTART: C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py ====
Traceback (most recent call last):
  File "C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py", line 4,
    in <module>
      prent ('c=', c)
NameError: name 'prent' is not defined
>>> |
Ln: 13 Col: 4
```

Логическая ошибка

```
proba.py - C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py (3.9.0)
File Edit Format Run Options Window Help
r = 5.4
c = 3.14 * r
s = 3.14 * r * r
print ('c=', c)
print ('s=', s)

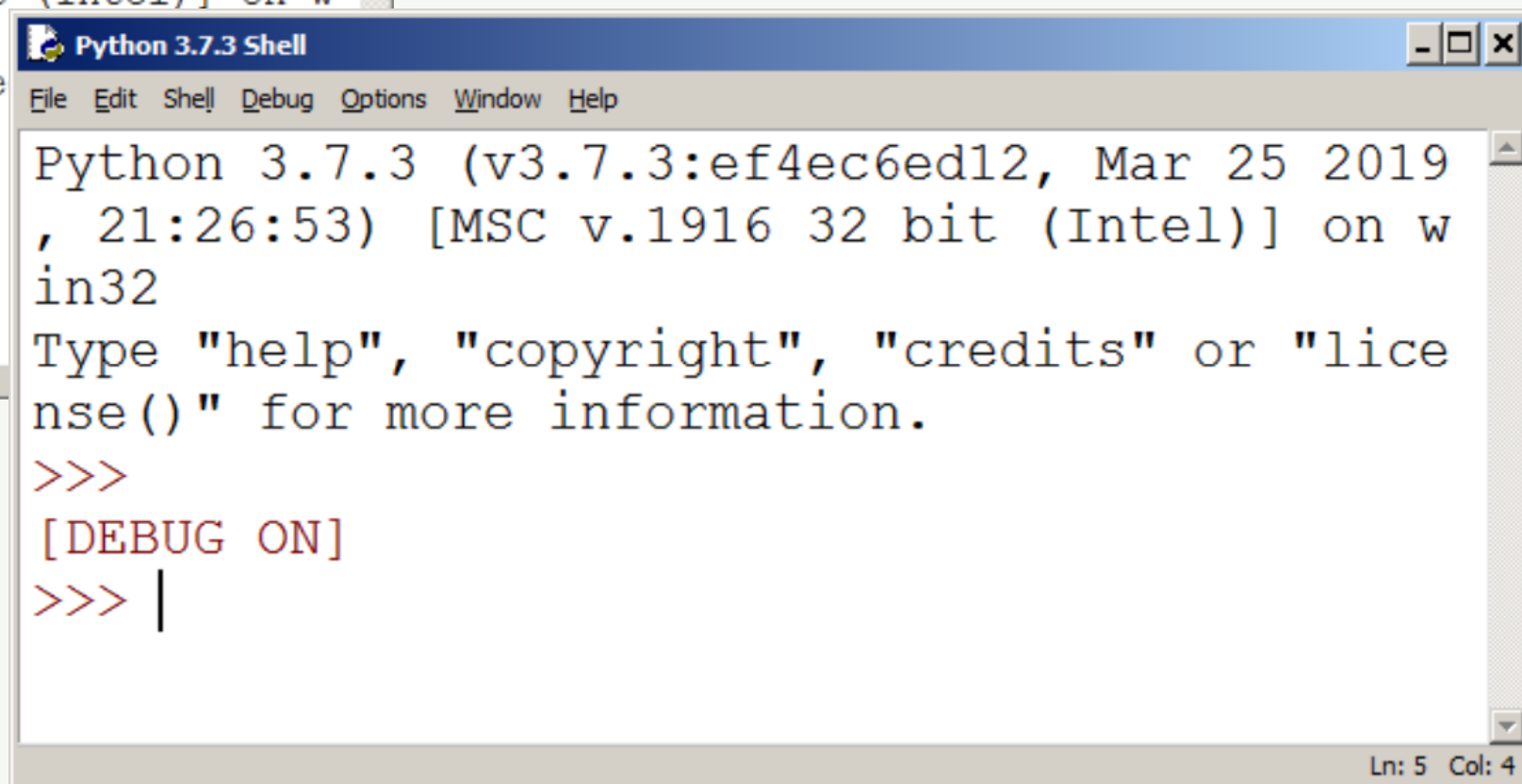
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
==== RESTART: C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py ====
c= 16.956000000000003
s= 91.56240000000003
>>> |
Ln: 17 Col: 4
```

ОТЛАДЧИК



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019
, 21:26:53) [MSC v.1916 32 bit (Intel)] on w
in32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

The image shows a Python 3.7.3 Shell window. A context menu is open over the 'Debug' menu item, with the following options: 'Go to File/Line', 'Debugger' (highlighted), 'Stack Viewer', and 'Auto-open Stack Viewer'. The shell prompt is '>>> |'.

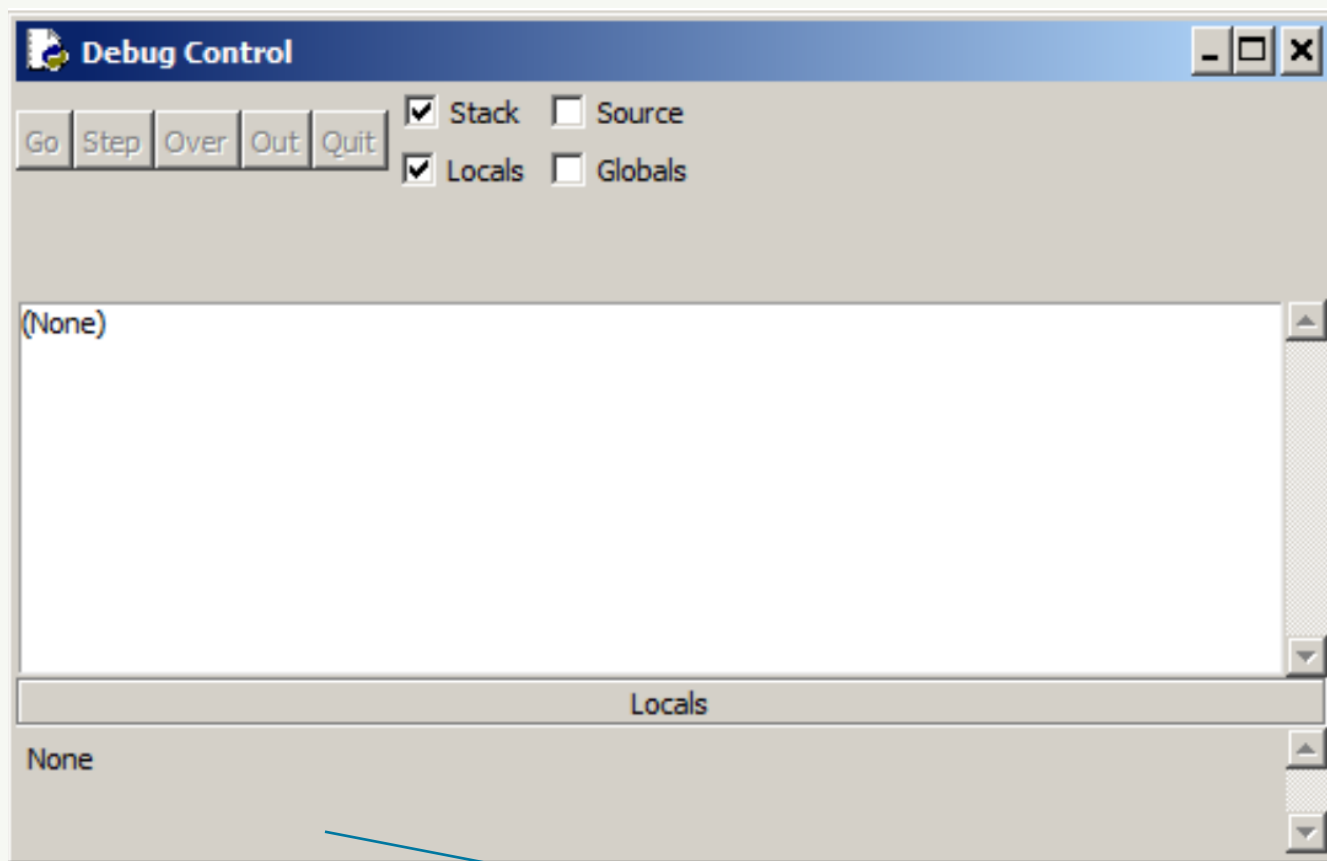


```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019
, 21:26:53) [MSC v.1916 32 bit (Intel)] on w
in32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
[DEBUG ON]
>>> |
```

The image shows a second Python 3.7.3 Shell window. The output of the 'Debugger' option is shown as '[DEBUG ON]' in red text. The shell prompt is '>>> |'. The status bar at the bottom right of the window shows 'Ln: 5 Col: 4'.



ОКНО ОТЛАДЧИКА



Go – непрерывное выполнение программного кода;

Step – построчное выполнение программного кода;

Over – переход к результату выполнения цикла или функции, если нет необходимости выполнять их построчно;

Out – приостановка выполнения программного кода после завершения выполнения цикла или функции;

Quit – остановка работы отладчика на некоторое время.

Панель для отслеживания текущих значений переменных

ТОЧКА ОСТАНОВА

Точка останова позволяет непрерывно выполнять программный код до того места, где предположительно может находиться ошибка.

Установка точки останова на определённой строке кода - в щелчок правой кнопкой мыши и выбор в контекстном меню **Set Breakpoint**. Выбранная строка окрашивается жёлтым цветом.

Удаление точки - щелчок правой кнопкой мыши на строке и выбор в контекстном меню **Clear Breakpoint**.



КАК ПРОВЕСТИ РАСЧЕТЫ ПРИ ДРУГИХ ЗНАЧЕНИЯХ r ?

Программа находит длину окружности и площадь круга для единственного значения радиуса (5,4 см).

Вычисления для другого значения радиуса требуют внесения изменений в текст программы, что неудобно и не всегда возможно.

Существует специальный оператор для ввода исходных данных в процессе работы программы, не прибегая к изменению текста программы!



ВВОД ДАННЫХ С КЛАВИАТУРЫ

Ввод в оперативную память значений переменных:

```
<имя переменной> = input()
```

Переменная будет строкового типа (`str`)

Пара скобок говорит о том, что мы вызываем функцию. Их надо писать обязательно, даже если в скобках ничего нет.

Выполнение оператора **input**:

- 1) компьютер переходит в режим ожидания данных
- 2) пользователь вводит данные с клавиатуры
- 3) пользователь нажимает клавишу `Enter`
- 4) строка, набранная пользователем, записывается в переменную с указанным именем



ВВОД ДАННЫХ С КЛАВИАТУРЫ

Ввод переменных целого типа:

```
<имя переменной> = int (input())
```

Переменная из строковой сразу преобразуется в целую

Ввод переменных вещественного типа (float):

```
<имя переменной> = float (input())
```

Переменная из строковой сразу преобразуется в вещественную

Ввод переменных с приглашением:

```
n = int (input ("введи n:"))
```

ВВОД ДАННЫХ С КЛАВИАТУРЫ

Каждый оператор `input()` захватывает только одну строку данных, причём захватывает её целиком.

Для того чтобы ввести в одной строке несколько чисел, разделённых пробелом, используют функцию `split` (от англ. *split* – расщепить).

```
a, b = input().split()
```

Ввод двух строковых величин,
разделённых пробелом

```
a, b = int(a), int(b)
```

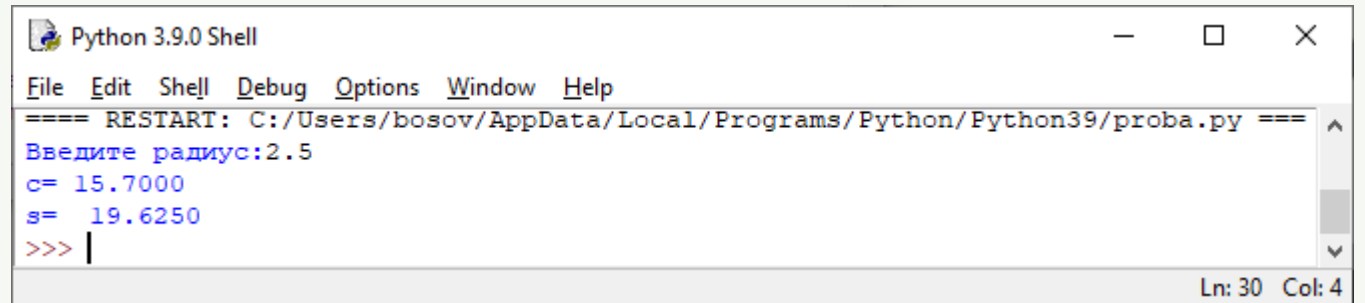
Преобразование к целому типу

Функция `map()` позволяет сократить запись считывания и преобразования нескольких считанных значений в числовой тип:

```
a, b, c = map(int, input().split())
```

УЛУЧШЕННАЯ ПРОГРАММА

```
r = float(input('Введите радиус:'))
c = 2 * 3.14 * r
s = 3.14 * r * r
print ("c=", "{:6.4f}".format(c))
print ("s=", " {:6.4f}".format(s))
```



Python 3.9.0 Shell

File Edit Shell Debug Options Window Help

==== RESTART: C:/Users/bosov/AppData/Local/Programs/Python/Python39/proba.py ====

Введите радиус:2.5

c= 15.7000

s= 19.6250

>>> |

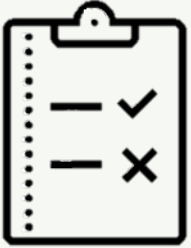
Ln: 30 Col: 4

Оператор ввода (функция) `input()` вводит с клавиатуры символьную строку. Для преобразования строки в целое число её обрабатывают функцией `int()`, для перевода в вещественное число – функцией `float()`.

Для вывода данных из оперативной памяти на экран монитора используется оператор вывода (функция) `print()`. Элементы списка вывода разделяются запятыми. По умолчанию при выводе данные разделяются пробелами.

Формат вывода – это указание количества знакомест на экране, которые должна занимать выводимая величина. Форматный вывод данных выполняется с помощью функции `format()`.

Ввод исходных данных и вывод результатов должны быть организованы понятно и удобно; это обеспечивает дружелюбность пользовательского интерфейса.



ВОПРОСЫ И ЗАДАНИЯ

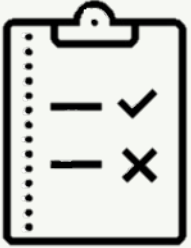
Что является результатом выполнения следующего оператора?

а) `print(a)`

б) `print('a')`

в) `print('a=', a)`



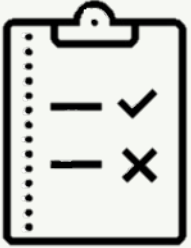


ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, выводящую на экран забавное изображение:

```
( \_ / )  
( = ' . ' = )  
( " ) _ ( " )
```





ВОПРОСЫ И ЗАДАНИЯ

Что будет выведено в результате работы следующей программы?

```
a = 1; b = 2; c = 3
```

```
print("{:3}".format(a))
```

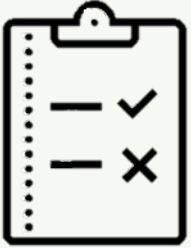
```
print("{:2}{:1}{}".format(b, b, b))
```

```
print("{}{}{}{}{}".format(c, c, c, c, c))
```

```
print("{:2}{:1}{}".format(b, b, b))
```

```
print("{:3}".format(a))
```





ВОПРОСЫ И ЗАДАНИЯ

Что будет выведено в результате работы следующей программы?

```
x = 143.511
```

```
print(x)
```

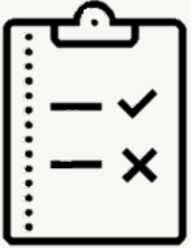
```
print("{:8.2f}".format(x))
```

```
print("{:.6f}".format(x))
```

```
print("{:10.3e}".format(x))
```

```
print("{:12.3e}".format(x))
```





ВОПРОСЫ И ЗАДАНИЯ

Определите результат работы программы, если переменным *a* и *b* были присвоены значения 2 и 4 соответственно.

```
a = int(input())
```

```
b = int(input())
```

```
a = a * a
```

```
b **= 2
```

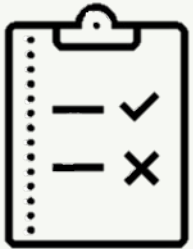
```
k = a * b
```

```
k *= 2
```

```
k += a + b
```

```
print(k)
```





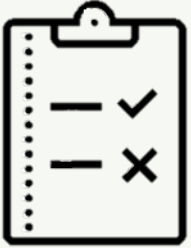
ВОПРОСЫ И ЗАДАНИЯ

Целочисленным переменным i , j , k нужно присвоить соответственно значения 10, 20 и 30.

Запишите оператор ввода, соответствующий входному потоку:

- а) 20 10 30
- б) 30 20 10
- в) 10 30 20



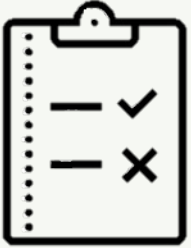


ВОПРОСЫ И ЗАДАНИЯ

Какой тип имеет переменная \underline{f} , если после выполнения оператора `print (f)` на экран было выведено следующее число?

- а) 125
- б) 125.0





ВОПРОСЫ И ЗАДАНИЯ

Дан фрагмент программы:

```
a = 10
```

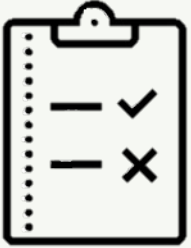
```
b = a + 1
```

```
a = b - a
```

```
print (a, b)
```

Какие числа будут выведены на экран компьютера?





ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, которая вычисляет площадь и периметр прямоугольника по двум его сторонам.



ВВОД ДАННЫХ В ОПЕРАТИВНУЮ ПАМЯТЬ

```
<имя переменной>=input("текст - подсказка")
```

СТРОКА

Приглашение

```
<имя переменной> = int(input())
```

ЦЕЛОЕ ЧИСЛО

```
<имя переменной> = float(input())
```

ВЕЩЕСТВЕННОЕ ЧИСЛО

```
print(<выражение 1>, < выражение 2>, ..., < выражение N>)
```

список вывода

ВЫВОД ДАННЫХ ИЗ ОПЕРАТИВНОЙ ПАМЯТИ НА ЭКРАН