**Ссылка на урок:** https://coreapp.ai/app/player/lesson/67e76548b5e585063c1fdb8b

(для учеников)

**Занятие 12**

**Тема**: **Реализация циклических алгоритмов на ЯП Python.**
**Цикл с параметром *For***

## 1. Теоретическая часть

На этом занятии мы рассмотрим ещё один вид циклического алгоритма – цикл с известным количеством повторений (цикл с параметром). Данный цикл записывается с помощью оператора **for** (в переводе с английского — «для»). Его общий вид:

**for** <параметр> **in** range (k, n, m):
  <операторы>

Здесь: <параметр> — переменная целого типа;

**range ()** — функция, описывающая необходимое количество повторов тела цикла, в скобках может быть указано от одного до трёх чисел:

• одно число (n) указывает на то, что нужно последовательно перебрать все целые числа от 0 до n - 1;

• два числа (к, п) говорят о том, что нужно последовательно перебрать все целые числа, находящиеся в диапазоне от k (начальное значение) до n - 1 (конечное значение);

• три числа (k, n, m) указывают на то, что параметр должен изменяться от k до n - 1 с *шагом*, равным m, **шаг** — это разница между следующим и предыдущим членами последовательности,

**например,** функция range (5) возвращает последовательность 0, 1, 2, 3, 4, функция range (2,8) возвращает последовательность 2, 3, 4, 5, 6, 7, функция range (0,22,3) возвращает последовательность 0, 3, 6, 9, 12, 15, 18, 21;

<операторы> — один или несколько операторов, составляющих тело цикла. О*ператоры тела цикла должны быть записаны с отступом.*

При выполнении цикла f*or* с тремя параметрами после каждого выполнения тела цикла происходит увеличение параметра цикла на шаг m. Если k ≥= n, цикл не выполнится ни разу.

Блок-схема работы цикла f*or* представлена на рисунке 1.


Рис. 1. Блок-схема цикла с параметром

## 2. Примеры программ с использованием цикла for

Рассмотрим примеры организации работы цикла с параметром.

Пример 1

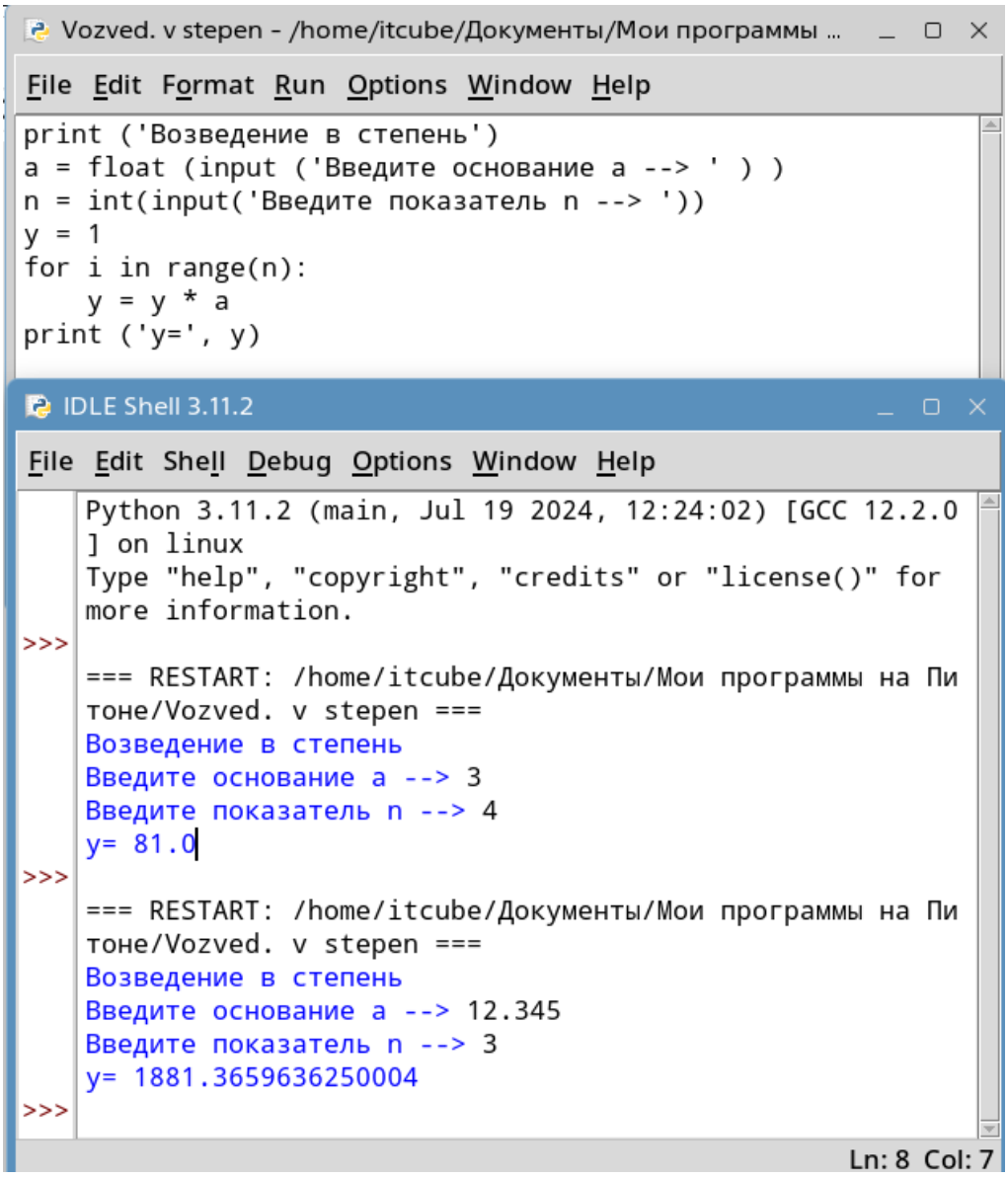
Составить программу вычисления степени с натуральным показателем **n** для любого вещественного числа **a**.

**Решение:**

```
print ('Возведение в степень')
a = float (input ('Введите основание a --> ' ) )
n = int(input('Введите показатель n → '))
y = 1 #первоначальное значение y полагаем равным единице
for i in range(n):
    y = y * a #операция умножения выполнятся в цикле ровно n раз
print ('y=', y) #вывод результата происходит вне цикла.
```

Примечание. Команда **print ('y=', y)** находится за пределами тела цикла, так как вывести на экран нужно только последнее значение переменной y (конечный результат) .

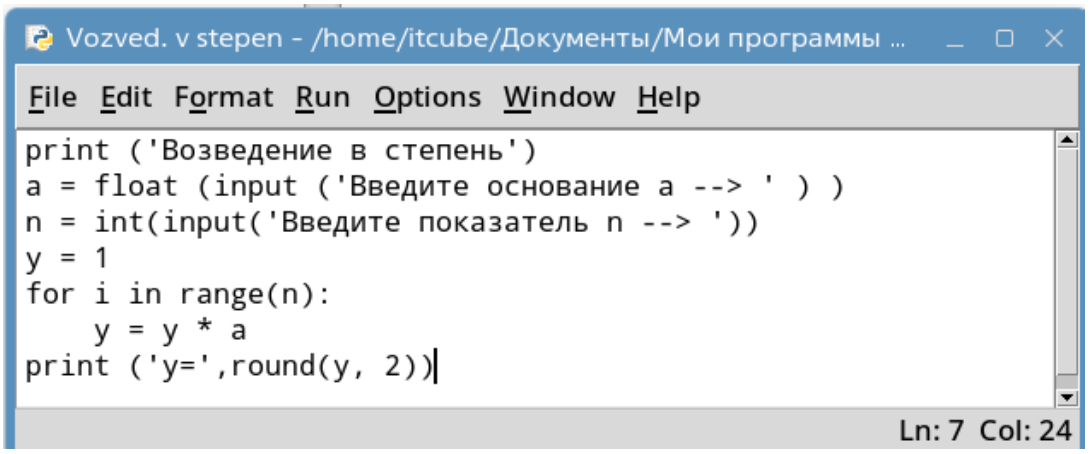
*Результат выполнения программы:*



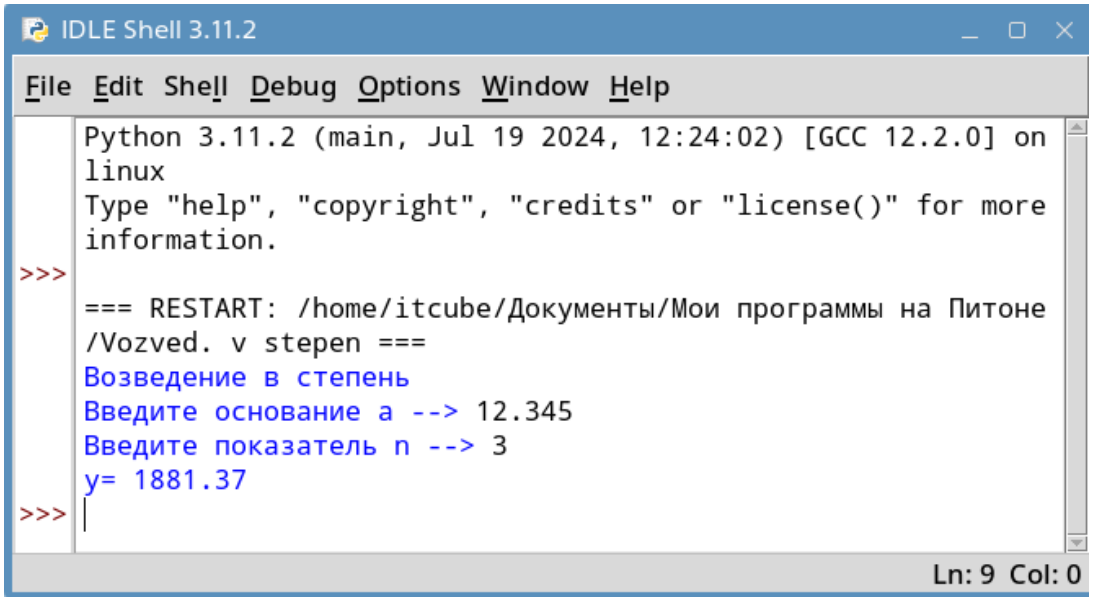
Обратите внимание, что при выводе результата возведения в степень вещественного числа 12,345 в его дробной части слишком много цифр, которые не имеют для нас большого значения. Давайте округлим результат до двух знаков после запятой. Мы уже знаем, что это можно сделать с помощью функции **round(),** которая округляет переданное ей число до количества указанных во втором параметре знаков, например: y=(3.756789067, 2); print(y)

# Результат: 3.76

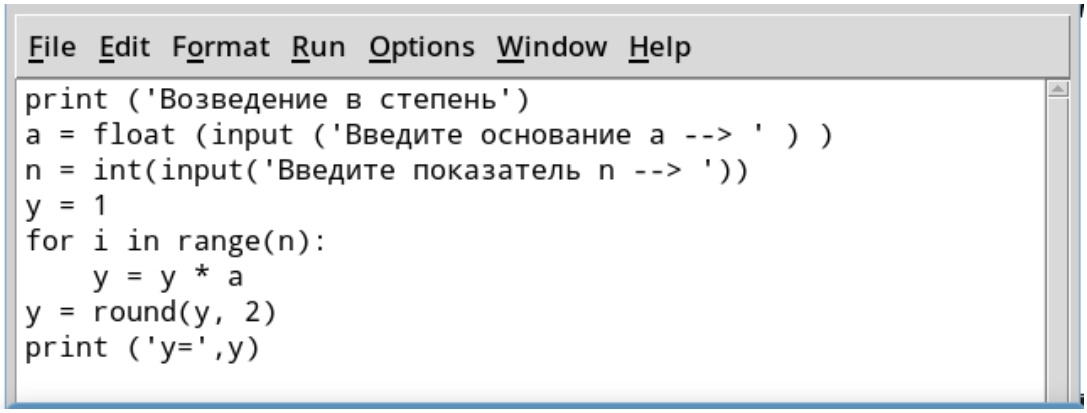
Внесём изменения в нашу программу




и выполним её:




Функцию round() мы встроили в оператор *print, для упрощения программы.* Но можно написать и так:




Результат всё-равно будет один и тот же (y= 1881.37).

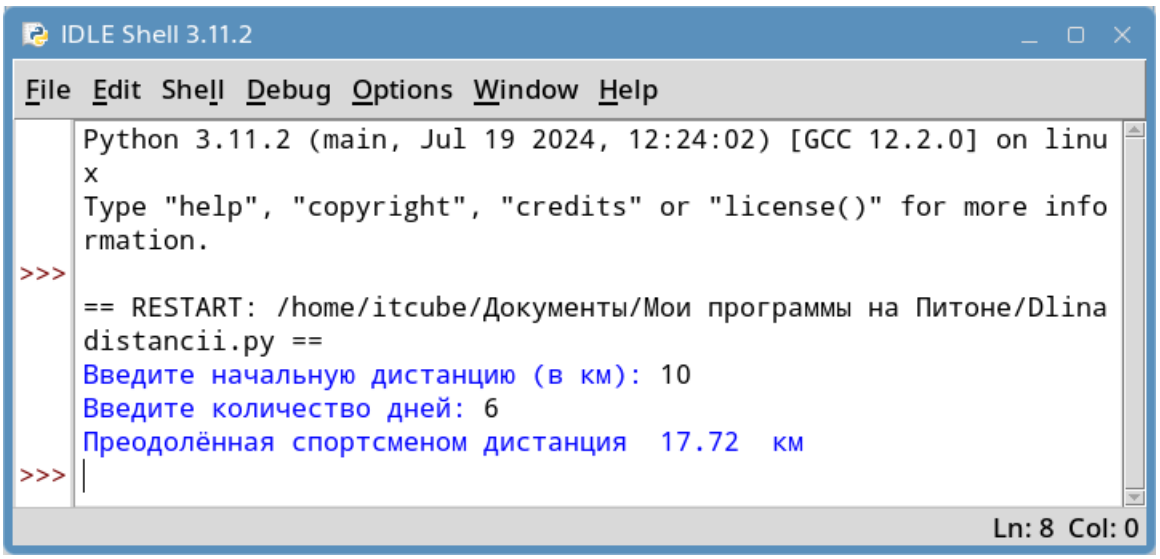
Пример 2

На тренировке спортсмен ежедневно пробегает некоторую дистанцию, с каждым днём увеличивая её на 10%. Написать программу, определяющую по расстоянию, преодолённому спортсменом в первый день тренировки, длину всей пройденной спортсменом дистанции с первого по k-й день.

**Решение:**

```
n = float(input("Введите начальную дистанцию "))
k = int(input("Введите количество дней "))
for i in range(k):
    n = n+n*0.1
print("Преодолённая спортсменом дистанция ", round(n, 2)," км")
```

*Результат работы программы:*




**Пример 3**

Напишите программу, которая выводит на экран таблицу степеней двойки (от нулевой до десятой). Рекомендуемый вид экрана после выполнения программы:

Таблица степеней двойки:

0 1
1 2
2 4
3 8
4 16
5 32
6 64
7 128
8 256
9 512
10 1024

**Решение:**

```
print (Таблица степеней двойки:')
for i in range(11):
    print(i,2**i)
```

Пример 4

Преобразуйте предыдущую программу так, чтобы выводить на экран таблицу степеней двойки (от нулевой до десятой) в десятичной и в двоичной системах счисления.

СПРАВКА

Алфавит *двоичной позиционной системыы счисления* состоит всего из двух цифр: 0 и 1. Количество цифр в алфавите называется *основанием* системы счисления и обозначается, чаще всего, буквой q. Пример двоичного числа: 00000101, что соответствует десятичному числу 5.

Метод деления на 2 основан на правиле перевода целых десятичных чисел в систему счисления с основанием q (школьный курс информатики, 8 класс).

Для перевода целого десятичного числа в двоичное нужно последовательно выполнять ***деление на 2*** сначала самого числа, а затем получаемых целых частных до тех пор, пока не получим частное, равное нулю. Причём частное подбираем так, чтобы в остатке от деления получались значения **0** или **1**.

Исходное число в двоичной системе счисления составляется последовательной записью полученных остатков от деления, начиная с последнего (то есть последний остаток будет первой цифрой двоичного числа).

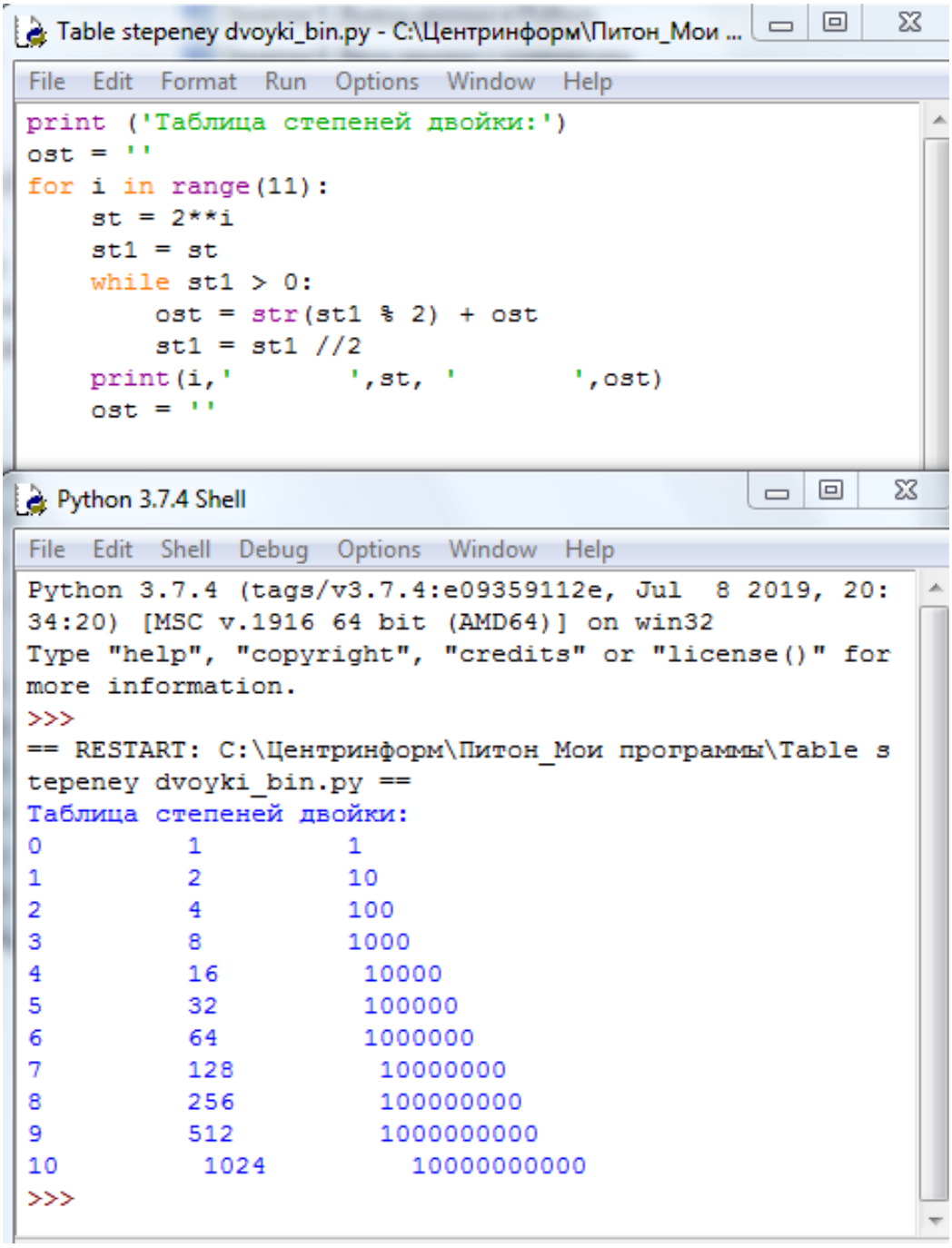
ПРИМЕР


**Решение:**

```
print ('Таблица степеней двойки:')
ost = '' #создаём переменную строкового типа для записи двоичного числа, с выводом без разделителей-пробелов
for i in range(11): #создаём цикл с параметром для вычисления степеней двойки
    st = 2**i
    st1 = st #переименовываем переменную st, чтобы использовать её во внутреннем цикле
    while st1 > 0: #создаём внутренний цикл с предусловием для представления степеней двойки в двоичном виде
        ost = str(st1 % 2) + ost #находим остаток от деления десятичного числа на 2,задаём ему тип – строка, складываем его с предыдущим остатком типа строка (то есть выполняем конкатенацию строк)
        st1 = st1 //2 #выполняем целочисленное деление получаемого частного на 2
```

```
print(i,'       ',st, '       ',ost) # печатаем результат в трёх столбцах, отделённых друг от друга 7 пробелами
ost = '' # очищаем переменную ost, подготавливая её для следующей итерации цикла
```

*Результат работы программы:*




Сравниваем результат с учебником:

| Число в десятичной СС | Степень числа 2 | Число в двоичной СС |
|---|---|---|
| 1 | $2^0$ | 1 |
| 2 | $2^1$ | 10 |
| 4 | $2^2$ | 100 |
| 8 | $2^3$ | 1000 |
| 16 | $2^4$ | 10000 |
| 32 | $2^5$ | 100000 |
| 64 | $2^6$ | 1000000 |
| 128 | $2^7$ | 10000000 |
| 256 | $2^8$ | 100000000 |
| 512 | $2^9$ | 1000000000 |
| 1024 | $2^{10}$ | 10000000000 |

При записи одного и того же алгоритма можно использовать как цикл **while,** так и цикл **for.** Это лишь вопрос целесообразности и предпочтений.

Примеры:

**Равносильные записи выражений**

```
d = 0
while d < 5 :
      d+=1
      print ( d )
```

```
for i in range(1,6):
    print ( i )
```

**Результат вывода на экран**

```
1
2
3
4
5
```

**Задача 2.** Выведем степени числа два от $2^1$ до $2^{10}$ (k = степени двойки).

**Равносильные записи выражений**

```
k = 1
while k <= 10 :
    print ( 2**k )
    k += 1
```

```
for k in range(1,11):
    print ( 2**k )
```

**3. Просмотр учебного видеоролика «Цикл for в Python»** (с целью расширения знаний, наглядного представления и закрепления материала):

https://rutube.ru/video/60f6ca1d6d5d7c1c7c61c21f8c7abff1/?r=plemwd (длина ролика: 09:16).

**4. Закрепление**

**ЗАДАНИЕ 1**. Запишите результат выполнения программы.

| 1) | 2) | 3) |
|---|---|---|
| `for i in range(10):`<br>`  if i==5:`<br>`  continue`<br>`  print(i*2,end=" ")` | `for i in range(10):`<br>`  if i==5:`<br>`  break`<br>`print(i*2, end=" ")` | `for c in range(0,22,3):`<br>`  print(c,end=" ")` |

**ЗАДАНИЕ 2**

Введите два целых числа A и B, не равных друг другу. Выполните следующие операции:
1) выведите все числа в диапазоне от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания, если A > B;
2) измените программу так, чтобы выводились только нечётные числа от A до B, в порядке возрастания, если A < B, или в порядке убывания, если A > B.

**!!!** Ответы к заданиям 1 и решение к заданию 2 смотрите на следующей странице. Но сначала попробуйте решить самостоятельно.

## ОТВЕТЫ

**ЗАДАНИЕ 1.**

1) 0 2 4 6 8 12 14 16 18
Пояснение. В данном примере при равенстве i==5 срабатывает оператор **continue**, в результате чего для i, равного 5, пропускается оператор print (i*2, end=" "). Поэтому число 10 не выводится на экран.

2) 0 2 4 6 8
Пояснение: В данном примере при равенстве i==5 срабатывает оператор **break**, в результате чего происходит завершение работы цикла. Значит, последнее значение **i**, рассмотренное в теле цикла, будет равно **4**.

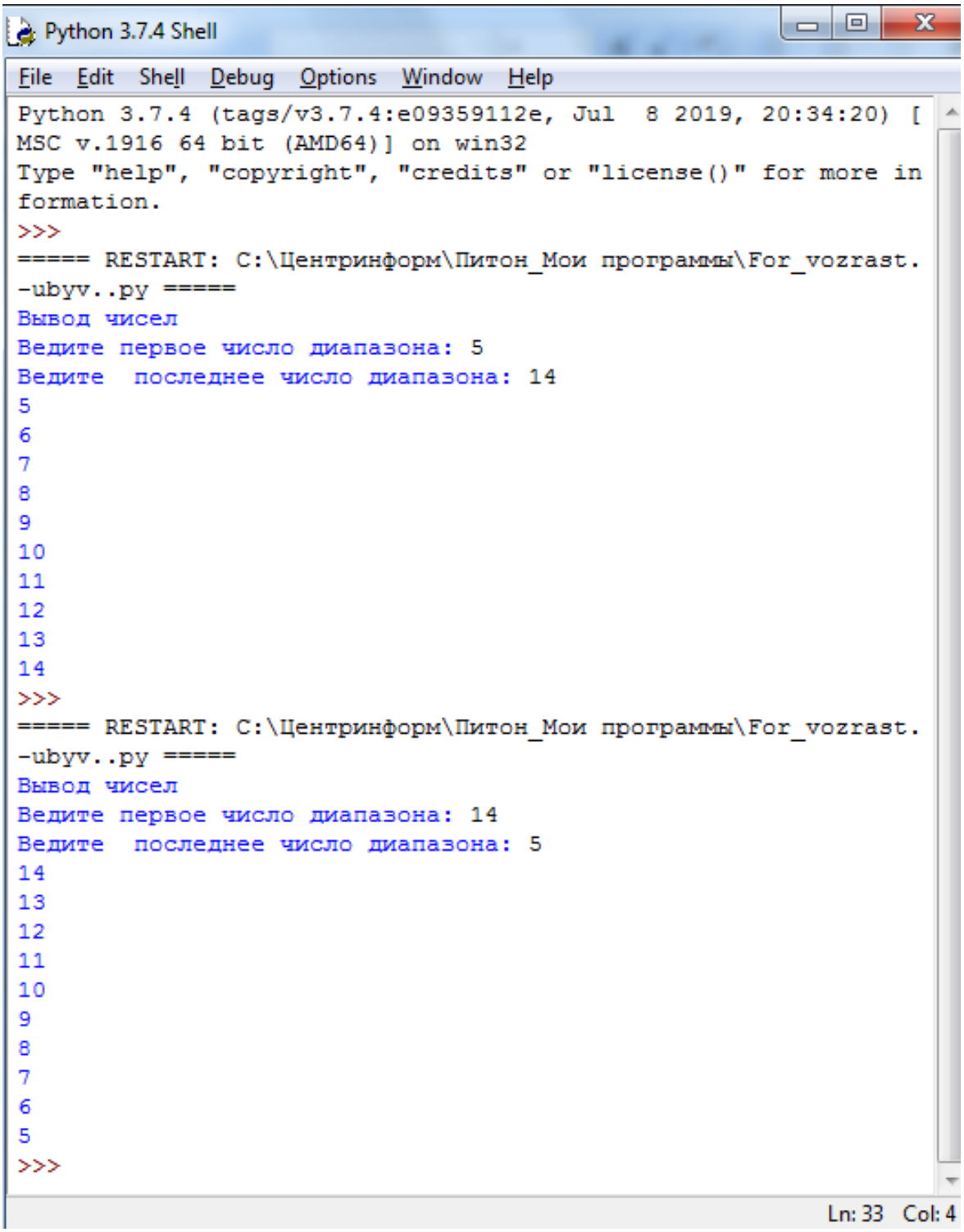
3) 0 3 6 9 12 15 18 21

**ЗАДАНИЕ 2**

**1)**
```
print ('Вывод чисел')
A = int(input('Введите первое число диапазона: '))
B = int(input ('Введите  последнее число диапазона: '))
if A < B:
    for x in range (A, B + 1):
        print (x)
else:
    for x in range (A, B - 1, -1): #цикл перебирает значения в обратном направлении
        print (x)
```

**2)**
```
print ('Вывод нечётных чисел')
A = int(input('Введите первое число диапазона: '))
B = int(input ('Введите  последнее число диапазона: '))
if A < B:
    for x in range (A, B + 1):
        if x % 2 != 0:  #если остаток от деления числа x на 2 не равен нулю, значит число нечётное
            print (x)
else:
    for x in range (A, B - 1, -1):
        if x % 2 != 0:
            print (x)
```

*1) Результат работы программы:*



*2) Результат работы программы:*