

Python. Lecture 02

Найдите все составные числа меньше N , которые представимы в виде произведения двух простых чисел.

```
#!/usr/bin/env python
```

```
def primes(N):  
    l = list(range(2,N))  
    for x in l:  
        n = x  
        while x*n <= N:  
            if x * n in l:  
                remove(x*n)  
            n+=1  
    return l  
  
def doublePrimes(N):  
    ps = primes(N)  
    out = []  
    for prime in ps:  
        for otherprime in ps[ps.index(prime):]:  
            doublePrime = prime*otherprime  
            if doublePrime < N:  
                out.append(tuple([prime, otherprime, doublePrime]))  
            else:  
                break  
    return out  
print "Prime multiples under 100"  
print doublePrimes(100)
```

Python

Строки...

Unicode & UTF-8

ЭТО РАЗНЫЕ ВЕЩИ!!!

Создание строк

- `'I am a string'`
- `"I too"`
- `'''Do not forget about me!'''`
- `"""I am a pretty multiline string!"""`
- `str([1, 2])`
- `str({'x': 1})`
- `"Don't forget about me!"`

Экранированные символы

- \\
- \'
- \"
- \n
- \t
- \uxxxx
- \Uxxxxxxxx

Сырые строки

r"Строка" – не экранируются символы

```
>>> s = "\t"
```

```
>>> print(s)
```

```
>>> s
```

```
'\t'
```

```
>>> s = r"\t"
```

```
>>> print(s)
```

```
\t
```

```
>>> s
```

```
'\\t'
```


Извлечение данных

```
>>> s = "It's interesting lecture!"
```

```
>>> "lecture" in s
```

```
True
```

```
>>> s.index("s")
```

```
3
```

```
>>> s.find("s")
```

```
3
```

```
>>> s.index("!!")
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in ?
```

```
ValueError: substring not found
```

```
>>> s.find("!!")
```

```
-1
```

«Изменчивость» строк

Строки не изменяемы!

```
>>> s = "It's interesting lecture!"
```

```
>>> s
```

```
"It's interesting lecture!"
```

```
>>> s[4]
```

```
''
```

```
>>> s[4]='_'
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in ?
```

```
TypeError: object doesn't support item assignment
```

Срезы

- `S = "Python"`
- `S[Start:Finish:Step]`
- `S[:]` #Python
- `"J"+S[1:]` #Jyton
- `S[:-1]` #Pytho
- `S[::-1]` #nohtyP

Форматирование строк

- “%s” % 10 # 10
- “%s - %s - %s” % (10, 20, 30)
- “%(x)s - %(b)s” % {“x” : 19, “b” : “Dad”}
- “%10d” % 2 # 2

Модификация

```
>>> s = " \n\t It's interesting lecture! \n\t\r"  
>>> s  
" \n\t It's interesting lecture! \n\t\r"  
>>> print(s)
```

It's interesting lecture!

```
>>> s.upper();  
" \n\t IT'S INTERESTING LECTURE! \n\t\r"  
>>> s.lower()  
" \n\t it's interesting lecture! \n\t\r"  
>>> s.lstrip()  
"It's interesting lecture! \n\t\r"  
>>> s.rstrip()  
" \n\t It's interesting lecture!"  
>>> s.strip()  
"It's interesting lecture!"
```

Модификация

Команды `strip`, `lstrip`, `rstrip`, `upper`, `lower` возвращают НОВУЮ строку.

НО!

```
>>> s = s.strip()
```

```
>>> s
```

```
"It's interesting lecture!"
```

Модификация

```
>>> xmltags = "<a><b>111</b>222</a>"
```

```
>>> xmltags.strip("<>");
```

```
'a<b>111</b>222</a'
```

```
>>> xmltags.strip("</a>");
```

```
'b>111</b>222'
```

```
>>> xmltags.strip("</ab>");
```

```
'111</b>222'
```

Извлечение данных

```
>>> s = "a,b,cccc,d"
```

```
>>> s.split(",");
```

```
['a', 'b', 'cccc', 'd']
```

```
>>> s.split(", ");
```

```
['a,b,cccc,d']
```

```
>>> s.split(",", 2);
```

```
['a', 'b', 'cccc,d']
```


Join

```
>>> some_list = ['one', 'two', 'three']
```

```
>>> ', '.join(some_list)
```

```
'one, two, three'
```

```
>>> ''.join(some_list)
```

```
'onetwothree'
```

```
>>> some_list2 = [1, 2, 3]
```

```
>>> ', '.join(some_list2)
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in ?
```

```
TypeError: sequence item 0: expected string, int found
```

```
>>> ', '.join([str(i) for i in some_list2])
```

```
'1, 2, 3'
```

Проверка типа содержимого

- S.isdigit()
- S.isalpha()
-
- S.istitle()

Unicode (Python 2)

```
>>> u"Привет"
u'\xf0\xd2\xс9\xd7\xс5\xd4'
>>> unicode("Привет", "koi8-r")
u'\u041f\u0440\u0438\u0432\u0435\u0442'
>>> s = unicode("Привет", "koi8-r")
>>> print s.encode("utf-8")
п÷я—пѠпѡ пѢ я Ѡ
>>> print s.encode("koi8-r")
Привет
```

Regexp

```
>>> import re
>>> regexp = "{{(. *?)}}"
>>> str = "{{this}} is {{strange}} string"
>>> for match in re.findall(regexp, str):
...     print "FIND: ", match
...
FIND: this
FIND: strange
```

Regex - compiled

```
>>> import re
>>> regexp = re.compile("{{(. *?)}}")
>>> str = "{{this}} is {{strange}} string"
>>> for match in regexp.findall(str):
...     print "FIND: ", match
...
FIND: this
FIND: strange
```

Regexp

- finditer
- match
- search

Чтение из файла

```
>>> file_in = open("test.txt", "r")
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
IOError: [Errno 2] No such file or directory: 'test.txt'
>>> file_in = open("foo.txt", "r")
>>> str = file_in.read()
>>> print str
Hello
i am
pretty
file!
>>> str.split()
['Hello', 'i', 'am', 'pretty', 'file!']
>>> str.splitlines()
['Hello', 'i am', 'pretty ', 'file!']
```

Запись в файл

```
>>> file_out = open("test.txt", "w")
>>> file_out.write("Test file\nNew line");
>>> file_out.close()
```

```
>>> try:
...     f = open("file.txt", "w")
...     f.write("test")
... finally:
...     f.close()
```


Работа с файлами файла - 2

- `read(size)`
- `readline(size)`
- `readlines(size)`

- `writelines`

Стандартный ввод и вывод

```
#!/usr/bin/env python
```

```
import sys
```

```
counter = 1
```

```
while True:
```

```
    line = sys.stdin.readline()
```

```
    if not line:
```

```
        break
```

```
    print "%s: %s" % (counter, line)
```

```
    counter += 1
```

Стандартный ввод

```
import sys
```

```
for I, line in enumerate(sys.stdin):  
    print "%s: %s" % (I, line)
```

```
sys.stdout.write("OK!")
```

StringIO

```
>>> from StringIO import StringIO
>>> str = StringIO("aaaa");
>>> str.read()
'aaaa'
>>> str.write("bbbb")
>>> str
<StringIO.StringIO instance at 0xb7d52acc>
>>> print str
<StringIO.StringIO instance at 0xb7d52acc>
>>> str.getvalue()
'aaaabbbb'
```

Urllib

```
>>> import urllib
```

```
>>> url_file = urllib.urlopen("http://spbau.ru")
```

```
>>> url_file.read(100)
```

```
'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-str'
```

```
>>>
```

ДЗ

1. Вывести греческий алфавит
2. Реализовать длинную арифметику (ЧЕСТНО!)
3. Используя модуль ElementTree, вывести в древовидном виде RSS ленту
4. Подсчитать на странице с результатами поиска Google статистику по доменам первого уровня