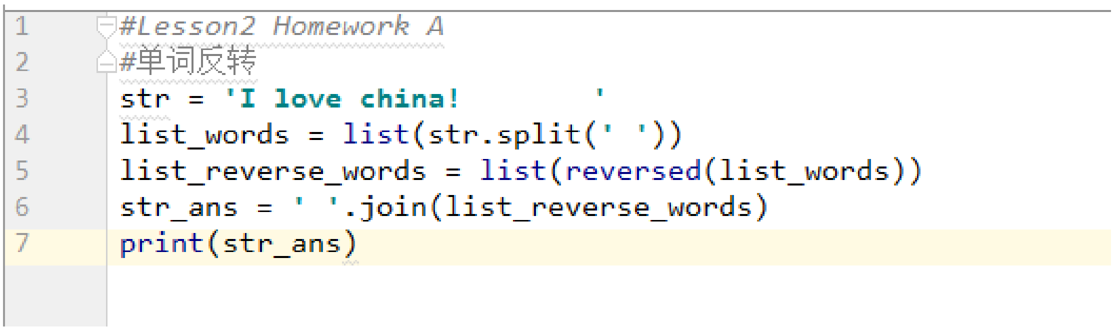
七月在线 python 基础第二节课作业

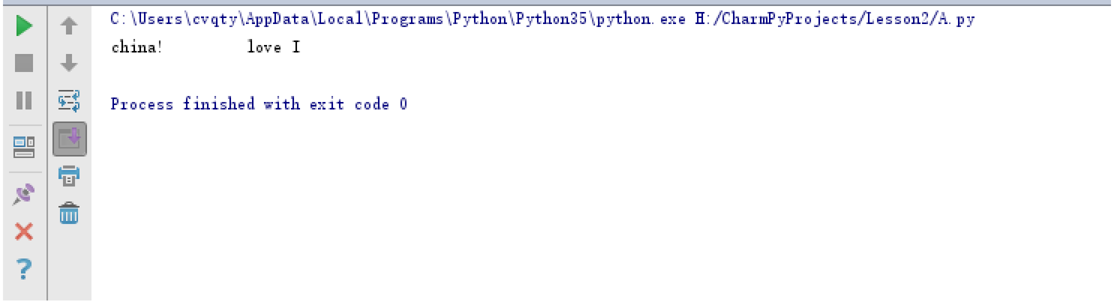
Author：粽子

2016/11/03

A．字符串按单词反转（必须保留所有空格）。' I love china!   '转化为"   china! love I '
代码如下：（可能现在有点问题）

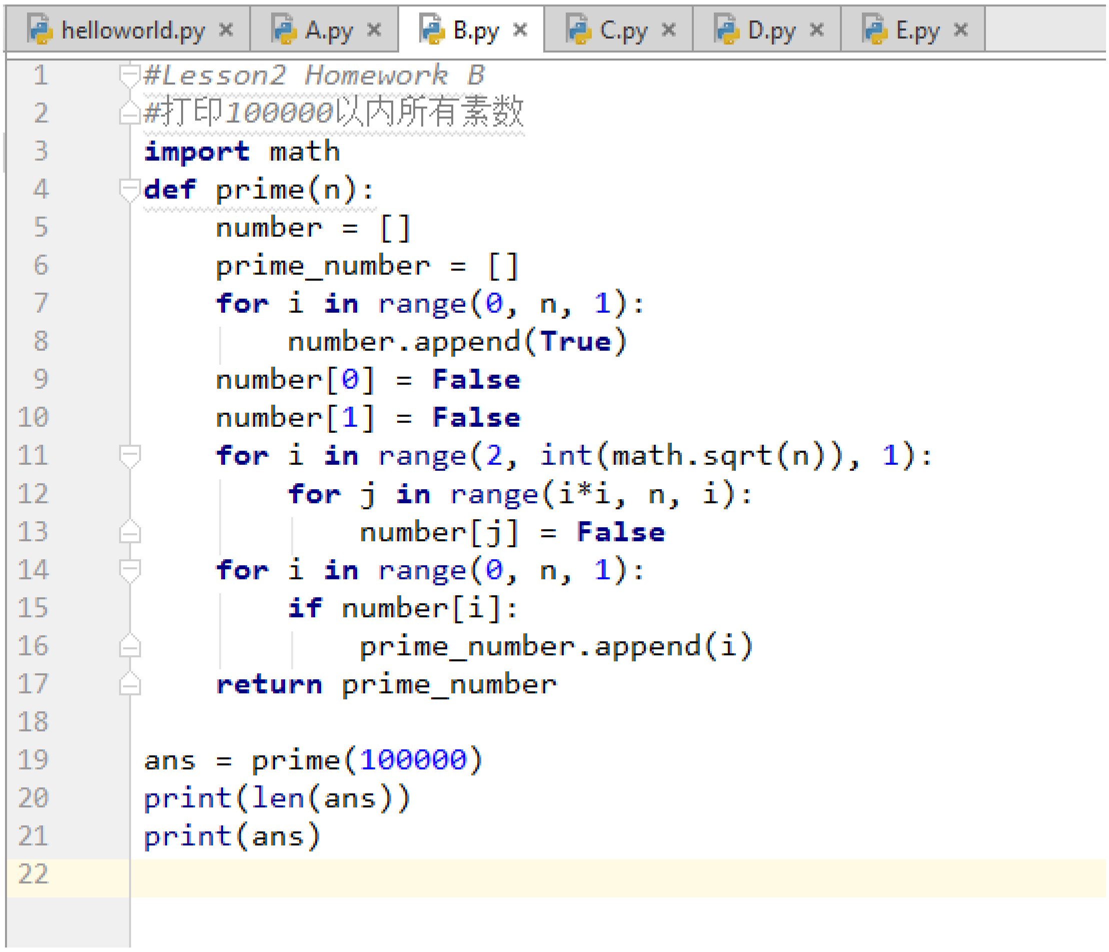


```python
#Lesson2 Homework A
#单词反转
str = 'I love china!          '
list_words = list(str.split(' '))
list_reverse_words = list(reversed(list_words))
str_ans = ' '.join(list_reverse_words)
print(str_ans)
```

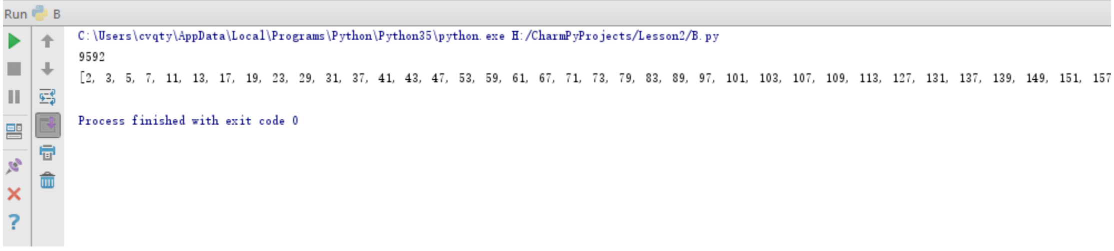


```
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/A.py
china!          love I

Process finished with exit code 0
```

B．打印 100000 以内的所有素数（筛法求素数）

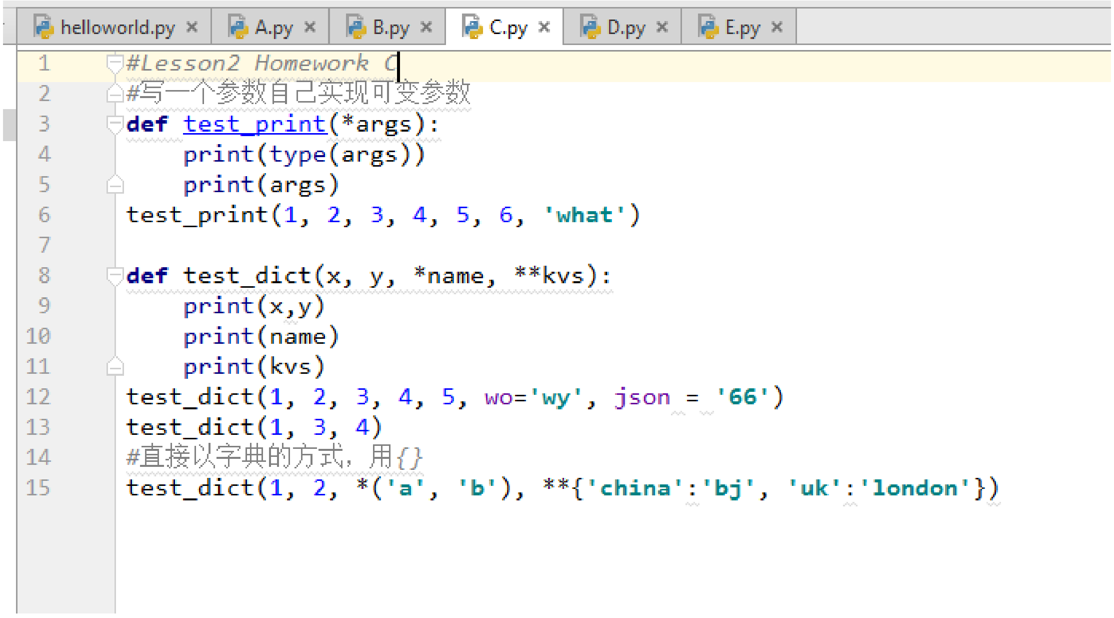


```python
#Lesson2 Homework B
#打印100000以内所有素数
import math
def prime(n):
    number = []
    prime_number = []
    for i in range(0, n, 1):
        number.append(True)
    number[0] = False
    number[1] = False
    for i in range(2, int(math.sqrt(n)), 1):
        for j in range(i*i, n, i):
            number[j] = False
    for i in range(0, n, 1):
        if number[i]:
            prime_number.append(i)
    return prime_number

ans = prime(100000)
print(len(ans))
print(ans)
```

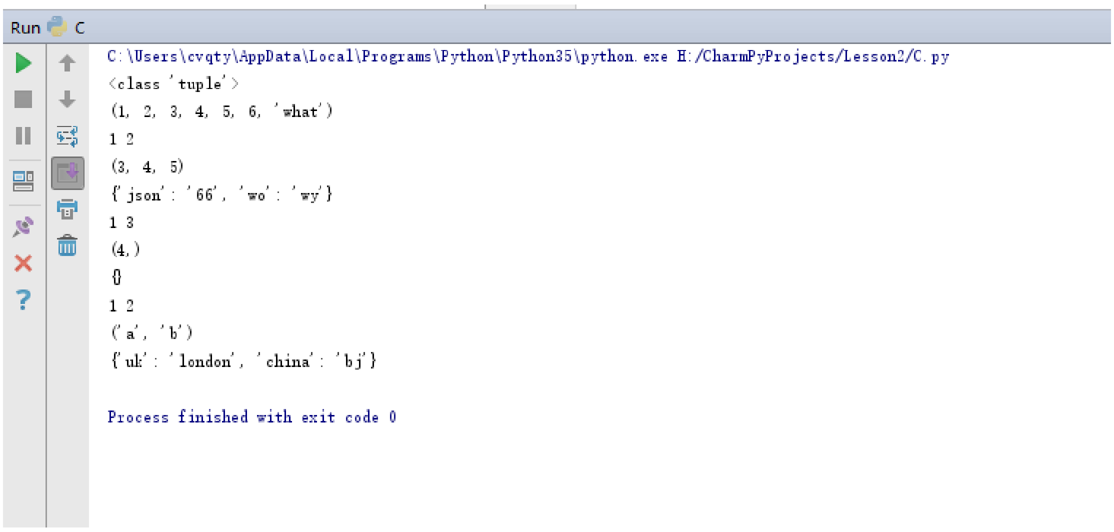


```
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/B.py
9592
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157

Process finished with exit code 0
```

C. 自己实现一个函数支持可变参数

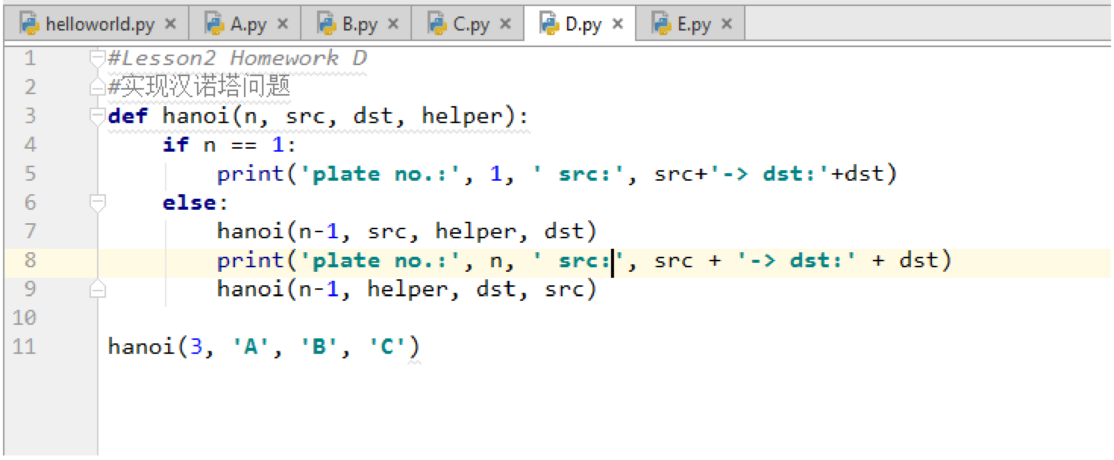


```
#Lesson2 Homework C
#写一个参数自己实现可变参数
def test_print(*args):
    print(type(args))
    print(args)
test_print(1, 2, 3, 4, 5, 6, 'what')

def test_dict(x, y, *name, **kvs):
    print(x,y)
    print(name)
    print(kvs)
test_dict(1, 2, 3, 4, 5, wo='wy', json = '66')
test_dict(1, 3, 4)
#直接以字典的方式，用{}
test_dict(1, 2, *('a', 'b'), **{'china':'bj', 'uk':'london'})
```

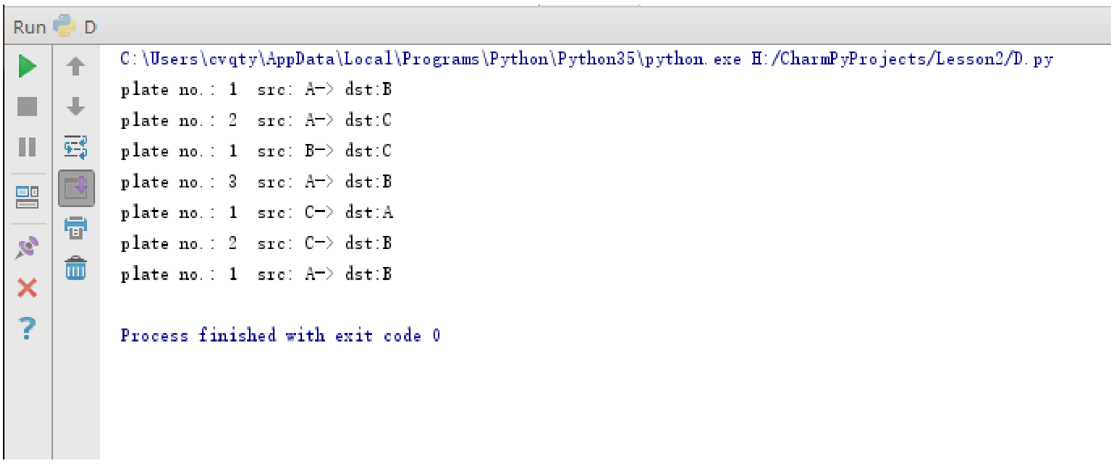


```
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/C.py
<class 'tuple'>
(1, 2, 3, 4, 5, 6, 'what')
1 2
(3, 4, 5)
{'json': '66', 'wo': 'wy'}
1 3
(4,)
{}
1 2
('a', 'b')
{'uk': 'london', 'china': 'bj'}

Process finished with exit code 0
```

D．自己实现函数解决 hanoi 塔问题



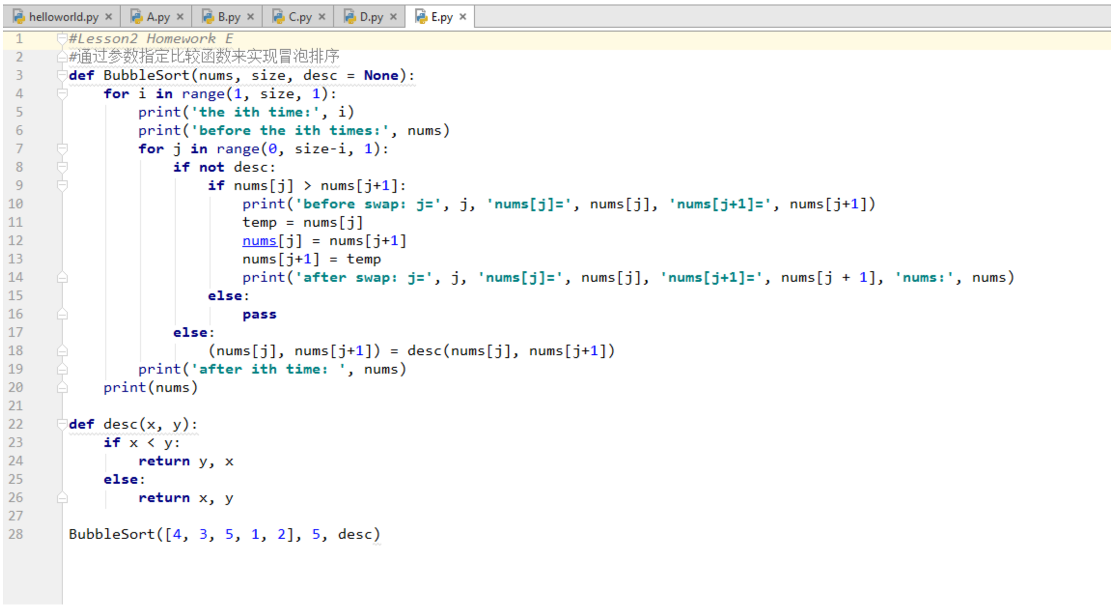
```
#Lesson2 Homework D
#实现汉诺塔问题
def hanoi(n, src, dst, helper):
    if n == 1:
        print('plate no.:', 1, ' src:', src+'-> dst:'+dst)
    else:
        hanoi(n-1, src, helper, dst)
        print('plate no.:', n, ' src:', src + '-> dst:' + dst)
        hanoi(n-1, helper, dst, src)

hanoi(3, 'A', 'B', 'C')
```



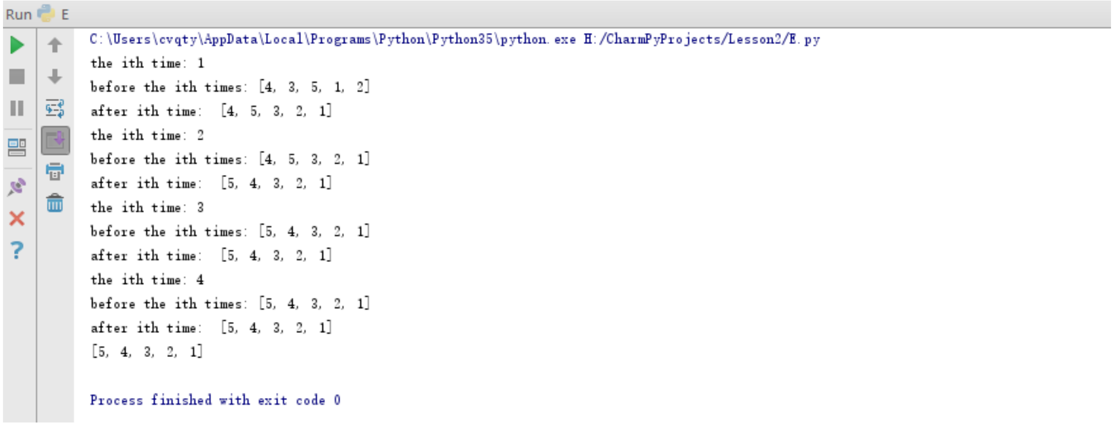
```
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/D.py
plate no.: 1  src: A-> dst:B
plate no.: 2  src: A-> dst:C
plate no.: 1  src: B-> dst:C
plate no.: 3  src: A-> dst:B
plate no.: 1  src: C-> dst:A
plate no.: 2  src: C-> dst:B
plate no.: 1  src: A-> dst:B

Process finished with exit code 0
```

E．实现一个 sort 函数，通过参数指定比较函数用来实现按不同顺序进行排序。

主要就用了冒泡排序，默认从小到大，如果指定比较参数就从大到小



```
#Lesson2 Homework E
#通过参数指定比较函数来实现冒泡排序
def BubbleSort(nums, size, desc = None):
    for i in range(1, size, 1):
        print('the ith time:', i)
        print('before the ith times:', nums)
        for j in range(0, size-i, 1):
            if not desc:
                if nums[j] > nums[j+1]:
                    print('before swap: j=', j, 'nums[j]=', nums[j], 'nums[j+1]=', nums[j+1])
                    temp = nums[j]
                    nums[j] = nums[j+1]
                    nums[j+1] = temp
                    print('after swap: j=', j, 'nums[j]=', nums[j], 'nums[j+1]=', nums[j + 1], 'nums:', nums)
                else:
                    pass
            else:
                (nums[j], nums[j+1]) = desc(nums[j], nums[j+1])
        print('after ith time: ', nums)
    print(nums)

def desc(x, y):
    if x < y:
        return y, x
    else:
        return x, y

BubbleSort([4, 3, 5, 1, 2], 5, desc)
```



```
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/E.py
the ith time: 1
before the ith times: [4, 3, 5, 1, 2]
after ith time:  [4, 5, 3, 2, 1]
the ith time: 2
before the ith times: [4, 5, 3, 2, 1]
after ith time:  [5, 4, 3, 2, 1]
the ith time: 3
before the ith times: [5, 4, 3, 2, 1]
after ith time:  [5, 4, 3, 2, 1]
the ith time: 4
before the ith times: [5, 4, 3, 2, 1]
after ith time:  [5, 4, 3, 2, 1]
[5, 4, 3, 2, 1]

Process finished with exit code 0
```