

练习题：

1、文件a1.txt内容

| 序号 | 部门 | 人数 | 平均年龄 | 备注 |
|----|--------|----|------|-----|
| 1 | python | 30 | 26 | 单身狗 |
| 2 | Linux | 26 | 30 | 没对象 |
| 3 | 运营部 | 20 | 24 | 女生多 |

.....

通过代码，将其构建成这种数据类型：

```
[{'序号': '1', '部门': 'Python', '人数': 30, '平均年龄': 26, '备注': '单身狗'}, .....
```

2、传入函数的字符串中,[数字]、[字母]、[空格] 以及 [其他]的个数，并返回结果。

3、写函数，接收两个数字参数，返回比较大的那个数字。

4、写函数，检查传入字典的每一个value的长度,如果大于2，那么仅保留前两个长度的内容，并将新内容返回给调用者。

```
dic = {"k1": "v1v1", "k2": [11,22,33,44]}
```

PS:字典中的value只能是字符串或列表

5、写函数，此函数只接收一个参数且此参数必须是列表数据类型，此函数完成的功能是返回给调用者一个字典，此字典的键值对为此列表的索引及对应的元素。例如传入的列表为：[11,22,33] 返回的字典为 {0:11,1:22,2:33}。

6、写函数，函数接收四个参数分别是：姓名，性别，年龄，学历。用户通过输入这四个内容，然后将这四个内容传入到函数中，此函数接收到这四个内容，将内容追加到一个student_msg文件中。

7、对第9题升级：支持用户持续输入，Q或者q退出，性别默认为男，如果遇到女学生，则把性别输入女。

8、写函数，用户传入修改的文件名，与要修改的内容，执行函数，完成整个文件的批量修改操作。

读代码，回答：代码中,打印出来的值a,b,c分别是什么？为什么？

```
a=10
b=20
def test5(a,b):
    print(a,b)
c = test5(b,a)
print(c)
```

9、读代码，回答：代码中,打印出来的值a,b,c分别是什么？为什么？

```
a=10
b=20
def test5(a,b):
    a=3
    b=5
    print(a,b)
c = test5(b,a)
print(c)
```

10、写函数,传入函数中多个实参(均为可迭代对象如字符串,列表,元祖,集合等),将每个实参的每个元素依次添加到函数的动态参数args里面.

例如 传入函数两个参数[1,2,3] (22,33)最终args为(1,2,3,22,33)

11、写函数,传入函数中多个实参(实参均为字典),将每个实参的键值对依次添加到函数的动态参数kwargs里面.

例如 传入函数两个参数{'name':'alex'} {'age':1000}最终kwargs为{'name':'alex' , 'age':1000}

12、下面代码成立么?如果不成立为什么报错?怎么解决?

题目一:

```
a = 2
def wrapper():
    print(a)
wrapper()
```

题目二:

```
a = 2
def wrapper():
    a += 1
    print(a)
wrapper()
```

题目三:

```
def wrapper():
    a = 1
    def inner():
        print(a)
    inner()
wrapper()
```

题目四:

```
def wrapper():
    a = 1
    def inner():
        a += 1
        print(a)
    inner()
wrapper()
```

13、写函数,接收两个数字参数,将较小的数字返回.

14、写函数,接收一个参数(此参数类型必须是可迭代对象),将可迭代对象的每个元素以'_'相连接,形成新的字符串,并返回.

例如 传入的可迭代对象为[1,'老男孩','武sir']返回的结果为'1_老男孩_武sir'

15、写函数,传入n个数,返回字典{'max':最大值,'min':最小值}

例如: 如:min_max(2,5,7,8,4) 返回: {'max':8,'min':2}(此题用到max(),min()内置函数)

16、写函数,传入一个参数n,返回n的阶乘

例如:cal(7) 计算7*6*5*4*3*2*1

17、写函数,返回一个扑克牌列表,里面有52项,每一项是一个元组

例如: [('红心', 2), ('草花', 2), ..., ('黑桃', 'A')]

18、有如下函数:

```
def wrapper():  
    def inner():  
        print(666)  
    wrapper()
```

你可以任意添加代码,用两种或以上的方法,执行inner函数.