信息处理技术 作业2

From: 梁鑫宇 3160104494

题目：对输入的汉字进行字频统计

思路分析：

1. 语言选择

由于题目涉及到对汉字的处理，并需要进行排序。c语言实现难度较大且可能效率较低，故考虑使用Python完成。

2. 数据类型

由于问题需要将汉字与出现的频次一一对应，考虑使用python中的字典，将汉字作为键，频次作为值。如此可以建立一个易于操作的对应关系。

同时又由于需要根据频次对汉字进行排序，考虑使用列表以方便地利用sort进行排序。于是为将汉字及频次一一对应地存储进相应的列表，建立一个类。

源代码：

```
# coding:gbk
# 定义一个类，使用它来存储汉字和对应的频次
class element:
    def __init__(self, x, y):
        self.character = x
        self.count = y

# 传入存储汉字及对应频次的字典和输入的汉字
def add_character(character,dic):
    # 如果输入的汉字在词典中尚未出现，则将之添加进去，并赋值为1
    if character not in dic:
        dic[character] = 1
    # 如果输入的汉字词典中已经已经包括，则将之对应的值+1即可
    else:
        dic[character] = dic[character] + 1

# 利用列表对汉字按频次进行排序
def show(dic):
    # 创建一个列表
    result = []
    for key in dic:
        # 将词典中的汉字和对应的频次存入以element类形式存入ele
        ele = element(key,dic[key])
        # 将存有汉字和对应频次的ele加入到result列表中
        result.append(ele)
    # 使用sort对列表进行排序
    result.sort(key=lambda element:element.count,reverse = True)
    # 将排序后的汉字及对应频次按顺序输出
```

```
    for ele in result:
        # 由于ele.count是int类型，不能与ele.character相加，所以强制转换成string再加和输出
        print(ele.character + ' : ' + str(ele.count))

def get_paragraph(dic):
    while True:
        try:
            # 此处用户以EOF的方式结束输入
            line = input()
        except:
            # 当用户结束输入，程序捕捉到EOF异常时结束
            break
        # 如果输入正常，那么将获得的输入交给函数add_character处理
        else:
            for character in line:
                # 此处判断是否为汉字
                if character >= u'\u4e00' and character <= u'\u9fa5':
                    add_character(character,dic)

dic = {}
get_paragraph(dic)
show(dic)
```

测试样例

测试文本：十九大报告全文