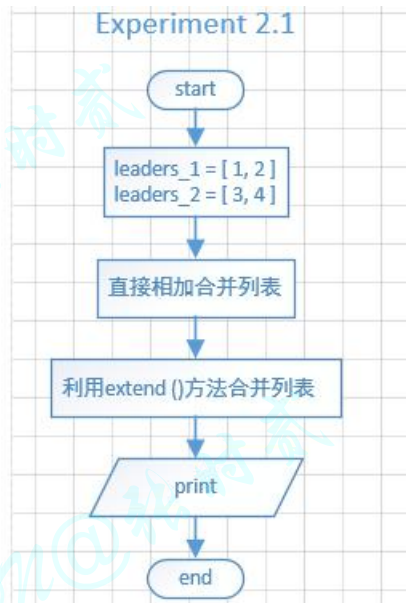


实验环境	严禁抄袭 仅供参考 Blog:zhangshier.vip
<b>一、实验题目</b> 实验二：组合数据	
<b>二、实验目的</b> <ol style="list-style-type: none"> <li>(1) 熟悉组合数据的类型。</li> <li>(2) 掌握列表、元组、字典、集合等组合数据的创建、访问方法。</li> <li>(3) 掌握组合数据推导式的使用方法。</li> <li>(4) 熟悉组合数据的常见应用。</li> </ol>	
<b>三、实验内容</b> <ol style="list-style-type: none"> <li>1、使用两种方法将两个列表中的数据合并。</li> <li>2、<code>set1={2, 5, 9, 1, 3}</code>, <code>set2={3, 6, 8, 2, 5}</code>, 调用集合操作符或函数完成以下功能：               <ol style="list-style-type: none"> <li>(1) 向 <code>set1</code> 中添加一个新的元素 7。</li> <li>(2) 求 <code>set1</code> 和 <code>set2</code> 的并集。</li> <li>(3) 求 <code>set1</code> 和 <code>set2</code> 的交集。</li> <li>(4) 求 <code>set1</code> 和 <code>set2</code> 的差集。</li> <li>(5) 判断给定关键字 <code>key=4</code> 是否在 <code>set1</code> 或 <code>set2</code> 中。</li> </ol> </li> <li>3、将某班学生的《Python 程序设计》这门课程的成绩保存在字典中,学号为键(key),分数为值 (value)。实现如下功能：               <ol style="list-style-type: none"> <li>(1) 向字典中添加学生成绩。</li> <li>(2) 修改字典中指定学生成绩。</li> <li>(3) 删除指定学生成绩。</li> <li>(4) 查询指定学生成绩。</li> <li>(5) 统计学生成绩,如最高分、最低分、平均分等。</li> </ol> </li> </ol>	
<b>四、实验步骤</b> <ol style="list-style-type: none"> <li>1、题目一               <ol style="list-style-type: none"> <li>(1) 问题分析 (含解决思路、使用的数据结构、程序流程图等) 列表相加,可以利用+号或者 <code>extend()</code> 方法</li> </ol> </li> </ol>	



(2) 算法/代码描述（基本要求源代码）

```

leaders_1 = [ 1, 2 ]
leaders_2 = [ 3, 4 ]
full_leaders_list = leaders_1 + leaders_2
print ( full_leaders_list )

```

```

leaders_1.extend ( leaders_2 )
print ( leaders_1 )

```

(3) 运行结果（含执行结果验证、输出显示信息）

```

"D:\太工\大三第二学期\Python\192062116 张帆\ve
[1, 2, 3, 4]
[1, 2, 3, 4]

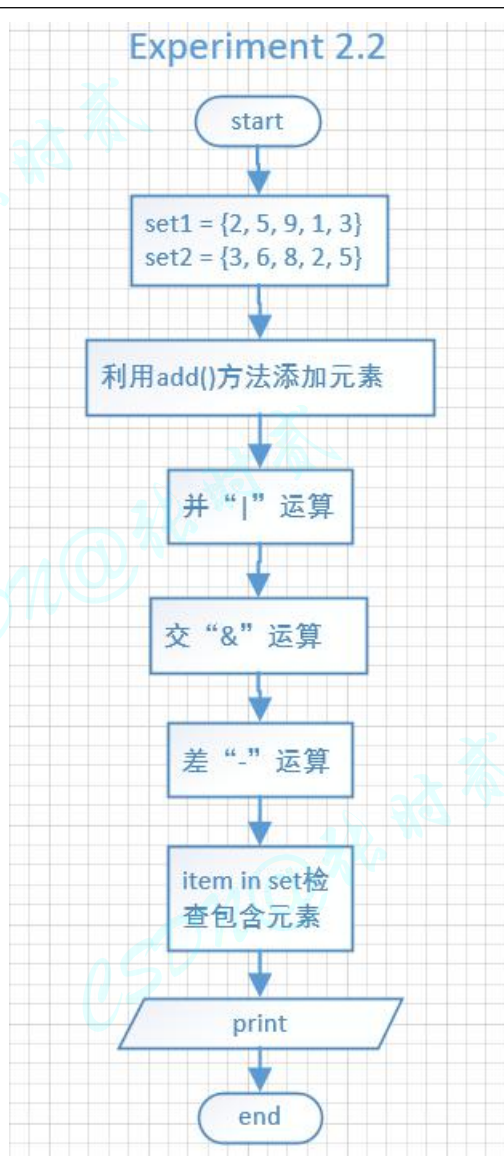
进程已结束，退出代码 0

```

## 2、题目二

(1) 问题分析（含解决思路、使用的数据结构、程序流程图等）

使用 `add()` 方法添加新元素，利用集合的运算符 `|`、`&`、`-`，做并交集运算，`item in set` 判断关键字



(2) 算法/代码描述（基本要求源代码）

```

set1 = {2, 5, 9, 1, 3}
set2 = {3, 6, 8, 2, 5}
set1.add ( 7 )
print ( "添加元素 7 后的集合为:", set1 )
print ( "集合 set1 和 set2 的并集为:", set1 | set2 )
print ( "集合 set1 和 set2 的交集为:", set1 & set2 )
print ( "集合 set1 和 set2 的差集为:", set1 - set2 )
print ( "关键字 key = 4 是否在集合中:", (4 in set1) or (4 in set2) )
  
```

(3) 运行结果（含执行结果验证、输出显示信息）

```
"D:\太工\大三第二学期\Python\192062116 张帆\venv\Script
添加元素7后的集合为: {1, 2, 3, 5, 7, 9}
集合set1和set2的并集为: {1, 2, 3, 5, 6, 7, 8, 9}
集合set1和set2的交集为: {2, 3, 5}
集合set1和set2的差集为: {1, 9, 7}
关键字key = 4 是否在集合中: False
```

### 3、题目三

(1) 问题分析（含解决思路、使用的数据结构、程序流程图等）

删除 `list.pop()`

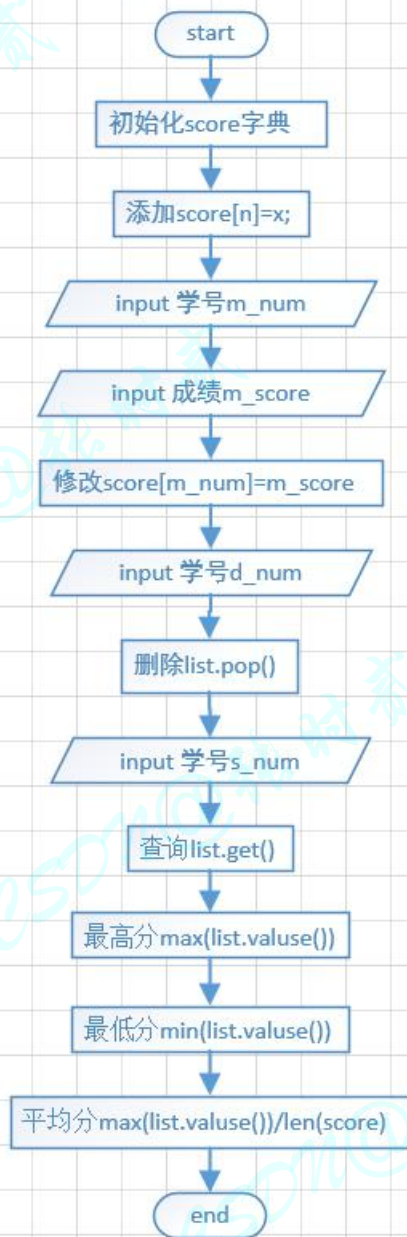
查询 `list.get()`

最高分 `max(list.valuse())`

最低分 `min(list.valuse())`

平均分 `max(list.valuse())/len(score)`

### Experiment 2.3



#### (2) 算法/代码描述（基本要求源代码）

```
score = {'001': 96, '002': 98, '003': 92, '004': 93, '005': 94}
print ("最初的学生成绩为:", score)
```

```
score['006'] = 100 # 添加
print ("向字典中添加 006 号学生成绩为:", score)
```

```
m_num = input ("修改成绩的学生学号")
m_score = int (input ("修改为"))
score[m_num] = m_score # 修改
print ("修改字典中学生成绩后:", score)
```

```

delete = input ( "输入删除学生学号" )
score.pop ( delete ) # 删除
print ( "删除学生成绩后:", score )

query = input ( "输入查询学生的学号" )
print ( "编号为 %s 的分数为: %d "%(query, score.get ( query ))) # 查询

print ( "最高分为:", max ( score.values () ) )
print ( "最低分为:", min ( score.values () ) )
print ( "平均分为:", sum ( score.values () )/len ( score ) )

```

### (3) 运行结果（含执行结果验证、输出显示信息）

```

"D:\太工\大三第二学期\Python\192062116 张帆\venv\Scripts\python.exe" "D:/太工/大三第二学期/Python/
最初的学生成绩为: {'001': 96, '002': 98, '003': 92, '004': 93, '005': 94}
向字典中添加006号学生成绩为: {'001': 96, '002': 98, '003': 92, '004': 93, '005': 94, '006': 100}
修改成绩的学生学号001
修改为90
修改字典中学生成绩后: {'001': 90, '002': 98, '003': 92, '004': 93, '005': 94, '006': 100}
输入删除学生学号001
删除学生成绩后: {'002': 98, '003': 92, '004': 93, '005': 94, '006': 100}
输入查询学生的学号002
编号为 002 的分数为: 98
最高分为: 100
最低分为: 92
平均分为: 95.4

```

## 五、 出现的问题及解决的方法

Input 默认输入时字符串，需要用 int 强转换，在 print 输出时，格式化字符是%s