

# 深圳大学实验报告

课程名称： 现代程序设计

实验项目名称： 复合数据类型

学院： 电子与信息工程学院

专业：

指导教师： 邹文斌

报告人：  学号：  班级：

实验时间： 2023 年 9 月 25 日

实验报告提交时间： 2023 年 9 月 27 日

教务部制

## 一、 实验要求

- a) 列表
- b) 程序流控制
- c) 需上交实验报告、py 源程序。

## 二、 实验环境

Python IDLE, Pycharm 等

## 三、 实验内容

### 1. 唯一元素的和

给定一个包含  $n$  个元素的列表，请找出列表中所有只出现一次的元素之和。

例如：

$n=6$ ，列表为  $[1, 2, 3, 1, 2, 4]$ ，

只出现一次的元素是 3 和 4，它们的和是 7。

**输入**

输入：

第一行一个正整数，为原列表的元素个数  $n$  ( $1 \leq n \leq 100$ )；

第二行，以字符串形式输入  $n$  个正整数(大小在  $1 \sim 10$  范围内)，相邻两个正整数用一个空格分隔。

提示：

```
s = input() #输入一个字符串，赋值给 s
```

```
a= s.split( ' ' ) #将字符串 s 按空格拆分，保存到列表 a 中
```

```
lis=[int(i) for i in a] #将列表 a 的元素转换成整数，保存到列表 lis 中
```

**输出**

一个正整数，表示列表中所有只出现一次的元素之和。

**输入样例 1**

6

1 2 3 1 2 4

**输出样例 1**

7

### 2. 组队

有  $n$  条蛇( $n$  为偶数)，准备两两组队参加“贪吃蛇巅峰对决”比赛，根据组委会的规定，只有长度相同的两条蛇才能组队参赛。

在比赛报名前，蛇可以吃金豆使自己变得更长。每条蛇吃 1 个金豆，长度会增加 1。

输入蛇的数量  $n$  和每条蛇的长度，请计算这些蛇至少要吃多少个金豆才能两两组队参赛。

例如：6 条蛇的长度分别为 5, 10, 2, 3, 14, 5，它们可以组成 3 队：

长度为 14 和 10 的蛇组成一队，需要吃 4 个金豆；

长度为 5 和 5 的蛇组成一队，需要吃 0 个金豆；

长度为 3 和 2 的蛇组成一队，需要吃 1 个金豆；

所以它们至少需要吃 5 个金豆。

#### 输入

第一行一个正整数，为蛇的数量  $n$  ( $2 \leq n \leq 20$ ，且  $n$  为偶数)：

第二行，按字符串形式输入  $n$  个正整数（大小在  $1 \sim 100$  范围内），相邻两个正整数用一个空格分隔，每个正整数表示一条蛇的长度。

#### 输出

一个整数，表示这些蛇要两两组队参赛，至少要吃多少个金豆。

#### 输入样例 1

```
6
5 10 2 3 14 5
```

#### 输出样例 1

```
5
```

### 3. 乘车系统

地铁的每站都有检票闸机，乘客进站和出站都要刷乘车卡，每刷一次卡，系统会记录持卡人的卡号（卡号不会有重号）。

地铁乘车系统根据刷卡的记录，可以计算出一段时间内的 3 项运营数据：

完成乘车的人次：进站和出站各刷一次卡，算作一次乘车，多次乘车计数累计；

正在乘车的人：进站后还没出站的乘客，计入正在乘车的人数；

乘客总人数：乘客人数指有刷卡记录的乘客总数，多次乘车不重复计数。已知一段时间内的全部刷卡记录，请分别计算完成乘车的人次、正在乘车的人数和乘客总人数。

例如：

系统获取的刷卡记录（卡号）是 4 8 4 2 11 4 2 4 8 4，其中，卡号为 8 的乘客乘车 1 次，已出站；卡号为 2 的乘客乘车 1 次，已出站；

卡号为 4 的乘客乘车 3 次，前 2 次已出站，第 3 次还没出站；卡号为 11 的乘客乘车 1 次，还没出站。

可知：

完成乘车的人次为 4，分别是：4（用卡号代表乘客）两次，2 和 8 各 1 次；

正在乘车的人数为 2，是 4 和 11；

乘客总人数为 4，分别是 2, 4, 8, 11。

#### 输入

一个字符串，包括若干个整数号码（整数的个数不超过 100），同一号码可能出现多次，相邻两个号码用一个空格分隔，表示一段时间内，系统记录的全部刷卡卡号记录。

#### 输出

三行，每行一行整数，分别是：这段时间内完成乘车的人次、正在乘车的人数和乘客总人数。

#### 输入样例 1

```
4 8 4 2 11 4 2 4 8 4
```

#### 输出样例 1

4

2

4

#### 4. 罗马数字

罗马数字起源于古罗马。

罗马数字的“数字”是七种字符：I、V、X、L、C、D 和 M, 这七种字符表示的数和阿拉伯数字表示的数，对照关系如下表：

罗马数字	I	V	X	L	C	D	M
阿拉伯数字	1	5	10	50	100	500	1000

罗马数字用几个字符的组合来表示更多的数。通常情况下，代表较小数的字符在代表较大数的字符右边。例如用罗马数字表示的数中：

2 写做 II, 即为两个 I;

12 写做 XII, 即为 X 加上 II;

27 写做 XXVII, 即为 XX 加上 V 加上 II。

罗马数字表示数，存在一些特例。比如 4 不写作 IIII, 而是 IV。字符 I 在字符 V 的左边，所表示的数等于较大数 5 减较小数 1 得到的数值 4。同样地，9 写作 IX。有 6 个数遵循以下特殊写法：

I 放在 V 的左边，表示 4;

I 放在 X 的左边，表示 9;

X 放在 L 的左边，表示 40;

X 放在 C 的左边，表示 90;

C 放在 D 的左边，表示 400;

C 放在 M 的左边，表示 900。

给定一个正整数，将其转为用罗马数字表示的数。

**输入：**

一个正整数  $x$ 。

**输出：**

一个字符串，表示  $x$  转换后的罗马数字。

**输入样例 1**

27

**输出样例 1**

XXVII

**输入样例 2**

43

**输出样例 2**

XLIII

## 四、实验过程

### 思路:

(涉及到算法实现的实验需阐述算法的逻辑关系)

#### 一、唯一元素的和

1. 首先键入数字，并将其保存在列表中。
2. 将列表中的数字（字符串）逐一转化为整型并保存在新列表中。
3. 用 `count()` 函数筛选出列表中只出现一次的元素。
4. 对只出现过一次的元素进行求和并打印结果。

#### 二、组队

1. 对输入的蛇的长度进行保存并生成可迭代对象，便于后续计算。
2. 对获得的列表进行排序。
3. 对相邻两个长度的蛇计算需要的金豆数，然后将这些金豆数加起来。
4. 打印结果。

#### 三、乘车系统

1. 输入刷卡记录，并用列表存储。
2. 遍历卡号并计数。
3. 根据刷卡次数计算完成乘车的人次（每两次刷卡算一次乘车）。
4. 根据刷卡次数计算正在乘车的人次（奇数次刷卡表示乘客还未出站）。
5. 输出完成乘车的人次、正在乘车的人次以及乘客总人数。

#### 四、罗马数字

1. 定义罗马数字的符号和对应的数值。
2. 定义一个字符串用于存储转换后的罗马数字。
3. 遍历符号和数值列表，当输入的数值大于等于当前数值时，将对应的符号添加到结果字符串中同时减去对应的数值。
4. 读取输入的正整数并转换为罗马数字。
5. 打印结果。

### 完整代码:

(必须有详细的注释)

### 一、 唯一元素的和 Sum\_of\_elements.py

```
n = int (input(' 请输入一个正整数 n (1<= n <=100):'))

getchar = input(' 请输入 {} 个大小范围在 1~10 之间的数字，相邻两个数字之
间用空格隔开： '.format(n))

#将输入的字符串保存到列表中

lst = getchar.split ( ' ')

list =[int(i) for i in lst]

#筛选出列表中只出现一次的元素

num =[]

for i in list:

    count = list.count ( i )

    if count ==1:

        num.append ( i )


#求和并打印

sum =0

for j in num:

    sum = sum + j

print("列表中所有只出现一次的元素之和为 {}".format(sum))
```

### 二、 组队 Snake\_beans.py

```
n = int(input("请输入蛇的数量 n (2<=n<=20，且 n 为偶数): "))

lengths = list(map(int, input("请输入 {} 个正整数，表示每条蛇的长度，相
邻两个数字之间用空格隔开： ".format(n)).split()))

beans = 0

#排序

lengths.sort()
```

```
#对每相邻两个进行计算
```

```
for i in range(1, n, 2):
```

```
    beans += lengths[i] - lengths[i-1]
```

```
print("这些蛇要两两组队参赛，至少要吃 {} 个金豆。".format(beans))
```

### 三、乘车系统 Ride\_system.py

```
card_numbers = input('请输入刷卡记录，各个卡号以空格分隔:').split()
```

```
card_counts = []
```

```
for card_number in card_numbers:
```

```
    found = False
```

```
    for card_count in card_counts:
```

```
        if card_count[0] == card_number:
```

```
            card_count[1] += 1 # 如果卡号已存在于列表中，增加计数器
```

```
            found = True
```

```
            break
```

```
    if not found:
```

```
        card_counts.append([card_number, 1]) # 如果卡号不存在于  
列表中，将其添加到列表并初始化计数器为 1
```

```
completed_trips_count = 0 # 初始化完成乘车的人次计数器
```

```
ongoing_trips_count = 0 # 初始化正在乘车的人次计数器
```

```
passenger_count = len(card_counts) # 计算乘客总人数为卡号的数量
```

```
for card_count in card_counts:
```

```
    count = card_count[1]
```

```
    completed_trips_count += count // 2 # 根据刷卡次数计算完成乘车的人次  
(每两次刷卡算一次乘车)
```

```
    ongoing_trips_count += count % 2 # 根据刷卡次数计算正在乘车的人
```

次（奇数次刷卡表示乘客还未出站）

```
print(completed_trips_count) # 输出完成乘车的人次
print(ongoing_trips_count) # 输出正在乘车的人次
print(passenger_count) # 输出乘客总人数
```

#### 四、罗马数字 Roman\_num.py

```
def int_to_roman(num):
    # 定义罗马数字的符号和对应的数值
    symbols = ["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX",
               "V", "IV", "I"]
    values = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]

    roman = "" # 用于存储转换后的罗马数字

    # 遍历符号和数值列表
    for i in range(len(symbols)):
        while num >= values[i]: # 当输入的数值大于等于当前数值时，将对应
            # 的符号添加到结果字符串中
            roman += symbols[i]
            num -= values[i] # 减去对应的数值

    return roman

# 读取输入的正整数
num = int(input('请输入一个正整数:'))

# 转换为罗马数字
roman_num = int_to_roman(num)
```



```
# 输出转换后的罗马数字
```

```
print(roman_num)
```

## 五、 实验结果

（运行结果，截图）

### 一、 唯一元素的和

```
== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task4\Sum_of_elements.py ==
请输入一个正整数 n (1<= n <=100):5
请输入5个大小范围在1~10之间的数字，相邻两个数字之间用空格隔开: 5 6 10 8 9
列表中所有只出现一次的元素之和为38
>>>
== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task4\Sum_of_elements.py ==
请输入一个正整数 n (1<= n <=100):7
请输入7个大小范围在1~10之间的数字，相邻两个数字之间用空格隔开: 3 1 3 5 7 5 1
列表中所有只出现一次的元素之和为7
```

### 二、 组队

```
>>>
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task4\Snake_beans.py
请输入蛇的数量n (2<=n<=20, 且n为偶数): 6
请输入6个正整数，表示每条蛇的长度，相邻两个数字之间用空格隔开: 2 1 6 2 7 3
这些蛇要两两组队参赛，至少要吃3个金豆。
>>>
```

### 三、 乘车系统

```
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task4\Ride_system.py ===
请输入刷卡记录，各个卡号以空格分隔: 4 5 4 16 8 5 5 3
2
4
5
```

### 四、 罗马数字

```
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task4\Roman.py
请输入一个正整数:998
CMXCVIII
>>>
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task4\Roman.py
请输入一个正整数:1024
MXXIV
```

## 六、 实验心得

（本次实验遇到的问题，解决过程，有什么收获等）

- 1.熟悉了列表的排序，元素索引和计数等操作。
- 2.学习了.format()函数的操作。
- 3.学习了 map()函数生成可迭代对象。
- 4.复习了 for 循环、while 循环、if 语句，加深了对条件设置的理解。
- 5.学习用 def 定义函数。

深圳大学学生实验报告用纸

成绩评定：

实验过程（60 分）	实验结果（30 分）	心得体会（10 分）	总分（100 分）

指导教师签字：                    年    月    日

备注：

- 注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
- 2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。