

深圳大学实验报告

课程名称： 现代程序设计

实验项目名称： 函数

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 11 月 6 日

实验报告提交时间： 2023 年 11 月 8 日

教务部制

一、 实验要求

- a) 函数
- b) 算法
- 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. 大家来“找茬”

某网站开发了一款“找错误”的小游戏。游戏提供两幅十分相似的图片，其中有许多细节部分不同。玩家各自在自己的屏幕上观察这两幅图，每发现一处“错误”，点击屏幕错误位置后，一条包含用户名和错误编号的信息会被系统记录下来。如果一处错误只有一个玩家发现，网站会奖励发现错误的玩家 100 积分；如果有两个以上的玩家都发现了同一处错误，发现错误的玩家平分这 100 积分的奖励(如果不能平分则向下取整)。

已知所有玩家共点击了 n 次屏幕，以及系统收集到的 n 条信息(用户名，错误编号)，输入一个用户名，输出他(她)在这场游戏中获得了多少积分。

例如：

$n=3$, 系统收集到的 3 条信息是：

A 4(用户 A 发现了编号为 4 的错误)

B 6(用户 B 发现了编号为 6 的错误)

A 6(用户 A 发现了编号为 6 的错误)

用户 A 共获得 150 积分。其中，编号为 4 的错误是 A 自己发现的，获得 100 积分；编号为 6 的错误 A 和 B 都发现了，各获得 50 积分。

输入

第 1 行一个正整数 n , 为所有玩家点击屏幕的数量($1 \leq n \leq 100$)；

第 2~第 $n+1$ 行，每行一个字符串，由用户名和错误编号组成(用户名是一个大写字母，错误编号是一个整数，用户名的种类不超过 26), 之间用一个空格分隔；

第 $n+2$ 行一个字符串，为指定的用户名。题目数据保证不会出现重复信息，并且指定的用户名存在于 n 条信息中。

输出

一个整数，为这名指定用户在这轮游戏中获得的积分。

输入样例 1

3

A 4

B 6

A 6

A

输出样例 1

150

2. 奖牌榜

今年大学生运动会, 需要根据每个学校获得的奖牌数量, 按奖牌总数从多到少进行排序, 整理出奖牌榜, 如果有相同总奖牌数, 则学校编号小的排在前面。

输入:

第 1 行是 1 个正整数, 表示参加运动会的学校数目 n ($1 \leq n \leq 2000$)

接下来的 n 行, 每行包含两个正整数, 分别为每个学校的编号 (不大于 999999) 和总奖牌数 (不大于 600), 中间用单个空格隔开。

输出:

n 行, 是将学校按总奖牌数从高到低的顺序进行排序的结果, 每行包含学校编号和总奖牌数, 之间用一个空格隔开。

输入样例:

```
3
1 40
2 53
3 53
```

输出样例:

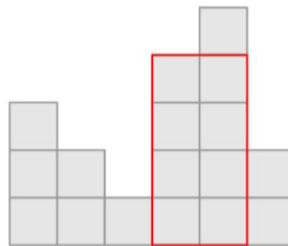
```
2 53
3 53
1 40
```

3. 求面积

一张半边参差不齐的网格纸 (网格边长均为 1), 有一边是完整没有破损的。现要从中剪出一片面积最大的矩形纸片。

给定网格纸中完整边的长度 N ($1 \leq N \leq 1000000$), 以及网格中每一列残存部分的高度 ($1 \leq \text{高度} \leq 10000$), 输出能够剪出的最大矩形纸片面积。

例如: $N=6$, 每一列残存部分的高度依次为 3、2、1、4、5、2, 如下图所示:



可以发现, 沿着红色框可以剪出的矩形纸片面积最大, 为 8, 所以输出 8。

输入描述:

第一行输入一个正整数 N ($1 \leq N \leq 1000000$), 表示纸片完整边的长度

第二行输入 N 个正整数 ($1 \leq \text{正整数} \leq 10000$), 表示每列格子残存部分的高度, 两个正整数之间用一个空格隔开。

输出描述:

输出一个正整数, 表示能够剪出的最大矩形纸片面积。

样例输入:

```
6
3 2 1 4 5 2
```

样例输出:

```
8
```

4. 公司活动 (A+挑战题)

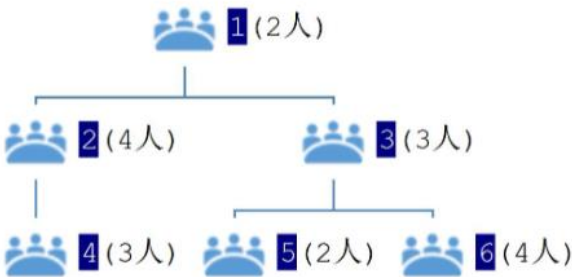
有一个公司由 N 个部门组成, 编号从 1 到 N 。这些部门之间的层次关系形成了一个树状结构, 一个上级部门可能会有 1 个或多个直接下级部门, 一个下级部门只有一个直接上级部门。

本月公司举办了一个大型活动, 这次的活动组织方按如下要求安排活动:

1. 来的人越多越好;
2. 如果一个上级部门参加本次活动, 那么他们的直接下级部门就不能参加, 而他的间接下集部门可以参加 (如下图, 如果部门 1 参加, 那么部门 2、3 不能参加, 而部门 4、5、6 可以参加)。

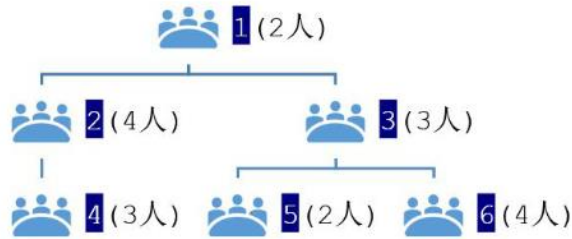
请你计算一下, 如何安排可以使参加活动的人数最多, 并输出参加活动的最多人数。

例如: 当 $N=6$, 每个部门编号为 1 到 6, 部门上下级关系和部门的人数如下图所示:



对应的数据为:

```
6
0 1 2
1 2 4
1 3 3
2 4 3
3 5 2
3 6 4
```



对应的数据为:

```
6
0 1 2
1 2 4
1 3 3
2 4 3
3 5 2
3 6 4
```

注意: 示例中, 部门 1 是层级最高的部门, 没有直接上级, 故将其直接上级部门设为 0; 当安排 (1、4、5、6) 这 4 个部门参加活动时, 人数最多, 为 11, 所以输出 11。

输入描述:

第一行输入一个正整数 N ($1 \leq N \leq 100000$), 表示公司所有部门的数量

接下来有 N 行, 每行输入三个非负整数 F 、 S 和 C , ($0 \leq F \leq N$, $1 \leq S \leq N$, $F \neq S$, $1 \leq C \leq 1000$), F 表示部门 S 的直接上级, C 表示部门 S 的人数, 整数之间用一个空格隔开。

注意: 如果是最上层的部门, 其直接上级部门编号设为 0

输出描述

输出一个整数, 表示参加活动的最多人数

样例输入:

```
6
0 1 2
1 2 4
1 3 3
2 4 3
```

3 5 2

3 6 4

样例输出:

11

5. 疏散行动 (A+挑战题)

为了避免太阳爆发引起的灾难，人类决定给地球装上发动机，最终逃离太阳系。原计划要带着月球一起走，结果月球行星发动机发生灾难性故障，必须炸毁月球。为此，在月球上的工作人员都要疏散回地球。月球基地有一艘太空穿梭机可以用来疏散工作人员。但是人们分散在各处，必须前往基地集合，他们到达基地的时间不等。穿梭机可以将抵达基地等待登机的工作人员先送回地球，然后再返回基地疏散下一批工作人员。

总共有 N 名工作人员需要疏散，太空穿梭机从月球到地球往返一次花时间 M 小时，第 i 个人抵达基地等待登机的时刻为 T_i 。

指挥官希望所有工作人员在基地等待的时间总和最小，而且他可以任意安排穿梭机的起飞时间，假定穿梭机足够大，可以装下所有工作人员，在不计登机和下机时间等因素的情况下，最小的等候时间总和是多少？

例如： $N=5$ ， $M=4$ ，1 号~5 号工作人员到达基地的时刻依次为 11、3、3、5、10，穿梭机可以在 3 时出发，先送 2 号、3 号工作人员去地球，然后于 7 时返回月球基地；

此时，4 号工作人员已于 5 时到达基地，等候了 2 小时。这时让穿梭机马上送走他，然后于 11 时从地球返回基地；

此时，5 号工作人员已于 10 时到达基地，等候了 1 小时；

而 1 号工作人员刚好于 11 时到达基地，等候 0 小时；

穿梭机于 11 时将两人送走，即完成全部疏散任务。总的等候时间=4 号工作人员等候时间+5 号工作人员等候时间=2+1=3 小时。

无法再找到有更小等候时间总和的方案。

输入描述:

第一行输入两个正整数 N ($1 \leq N \leq 500$)， M ($1 \leq M \leq 100$)，以一个空格隔开，分别表示工作人员人数和穿梭机的往返时间

第二行输入 N 个正整数，依次表示某个工作人员到达基地等候登机的时刻 T : ($1 \leq T \leq 4000000$)，相邻两数之间用一个空格隔开。

输出描述:

输出一个整数，表示所有工作人员等候时间之和的最小值(单位:小时)。

样例输入:

5 4

11 3 3 5 10

样例输出:

3

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、大家来“找茬”

1. 定义一个函数用于计算玩家获得的积分。
2. 遍历收集到的信息，记录错误编号的发现情况。
3. 根据错误编号的发现情况计算积分：如果只有一个玩家发现了该错误，给该玩家 100 积分；如果有多个玩家发现了该错误，则他们平分该错误的 100 积分。
4. 读取用户的输入，调用函数计算结果并输出。

二、奖牌榜

1. 获取用户输入，记录参加运动会的学校总数，初始化一个列表记录每个学校的编号和奖牌数。
2. 遍历输入每个学校的编号和奖牌数，并将数据加入到列表中进行统计。
3. 用 lambda 函数定义一个按奖牌数由高到低进行排序的函数并使用。
4. 用一个循环输出按照奖牌数由高到低进行排序后的结果。

三、求面积

1. 获取输入，并定义一个栈索引项。
2. 在循环中对每一列进行遍历，如果当前高度小于栈索引对应的高度，则可以计算栈项高度为高的矩形面积。
3. 弹出索引项，计算结束后更新最大面积的值并将当前索引放入栈。
4. 遍历完所有列后，如果还有剩余项，说明这些列还能继续计算，进行矩形面积的计算并更新最大值。
5. 得到计算结果，打印输出。

完整代码:

(必须有详细的注释)

一、大家来“找茬” find_errors

```
def calculate_score(n, data, username):  
    scores = {} # 记录错误编号被哪些玩家发现  
    user_scores = {} # 记录每个玩家的积分  
  
    # 遍历收集到的信息，记录错误编号的发现情况
```

```
for i in range(n):
    user, error = data[i].split()
    error = int(error)

    if error not in scores:
        scores[error] = []

    scores[error].append(user)

# 根据错误编号的发现情况计算积分
for error, users in scores.items():
    num_users = len(users)

    # 如果只有一个玩家发现了该错误, 给该玩家 100 积分
    if num_users == 1:
        user_scores[users[0]] = user_scores.get(users[0], 0) + 100

    # 如果有多个玩家发现了该错误, 平分 100 积分
    else:
        shared_score = 100 // num_users

        for user in users:
            user_scores[user] = user_scores.get(user, 0) + shared_score

return user_scores[username]

# 读取输入
n = int(input("请输入所有玩家点击屏幕的数量:"))
data = []

for j in range(n):
    data.append(input("请输入第 {} 条找到的错误, 由玩家名称和错误编号组成, 空格分隔:".format(j+1)))
```

```
username = input("请输入要统计积分总数的玩家名称:")
```

```
# 调用函数计算积分
```

```
score = calculate_score(n, data, username)
```

```
# 输出结果
```

```
print("该玩家的总分为:", score)
```

二、奖牌榜 cal_score.py

```
n = int(input("请输入学校总数量:")) #参加运动会的学校总数
```

```
#用于记录每个学校的编号和奖牌数
```

```
medal_counts = []
```

```
#输入每个学校的编号和奖牌数
```

```
for i in range(n):
```

```
    school, medal_count = map(int, input("请输入第{}组数据，以空格分隔学  
校编号和奖牌数:".format(i+1)).split())
```

```
    medal_counts.append((school, medal_count))
```

```
#按奖牌数由高到低进行排序
```

```
medal_counts.sort(key = lambda x: (-x[1], x[0]))
```

```
#输出结果
```

```
print("按照奖牌数由高到低进行排序后的结果为:")
```

```
for school, medal_count in medal_counts:
```

```
    print(school, medal_count)
```

三、求面积 max_area.py


```
N = int(input("请输入纸片完整边的长度:"))

heights = list(map(int, input("请输入每列格子残存部分的高度，用空格隔
开:").split()))

max_area = 0
stack = [] #用于记录高度的索引


for i, height in enumerate(heights):
    while stack and height < heights[stack[-1]]:
        #当前高度小于栈索引对应的高度，计算栈项高度为高的矩形面积
        top = stack.pop() #弹出索引

        #计算矩形的宽度
        width = i if not stack else i - stack[-1] - 1

        #计算矩形的面积
        area = heights[top] * width

        #更新最大面积
        max_area = max(max_area, area)

    stack.append(i) #将当前索引放入栈


while stack:
    #遍历完所有列后，如果还有剩余项，说明这些列还能继续计算
    top = stack.pop() #弹出栈索引

    #计算矩形的宽度
    width = N if not stack else N - stack[-1] - 1

    #计算矩形的面积
    area = heights[top] * width
```

```
#更新最大面积

max_area = max(max_area, area)


#输出结果

print("能够剪出的最大矩形面积为:", max_area)
```

五、 实验结果

(运行结果，截图)

一、大家来“找茬” find_errors

```
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task9\find_e
请输入所有玩家点击屏幕的数量:3
请输入第1条找到的错误，由玩家名称和错误编号组成，空格分隔:A 4
请输入第2条找到的错误，由玩家名称和错误编号组成，空格分隔:B 6
请输入第3条找到的错误，由玩家名称和错误编号组成，空格分隔:A 6
请输入要统计积分总数的玩家名称:A
该玩家的总分为: 150
>>>
```

二、奖牌榜

```
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task9\cal_score.py
请输入学校总数量:3
请输入第1组数据，以空格分隔学校编号和奖牌数:1 40
请输入第2组数据，以空格分隔学校编号和奖牌数:2 53
请输入第3组数据，以空格分隔学校编号和奖牌数:3 53
按照奖牌数由高到低进行排序后的结果为:
2 53
3 53
1 40
>>> |
```

三、求面积

```
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task9\max_area.p
请输入纸片完整边的长度:6
请输入每列格子残存部分的高度，用空格隔开:3 2 1 4 5 2
能够剪出的最大矩形面积为: 8
>>> |
```

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.增强了对字典对象的索引和处理能力。
- 2.练习使用了 `lambda` 函数。
- 3.复习运用了统计的各类方法。
- 4.初步学习了栈的思想并进行运用。
- 5.学会了 `for` 循环与 `while` 循环的结合使用。

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。