

# 图灵编程 12 月 1 日普及组月赛

时间：2024 年 12 月 1 日 18:20 ~ 21:30

题目名称	收尾人	委托	爪牙	武器
题目类型	传统型	传统型	传统型	传统型
可执行文件名	gloam.exe	morning.exe	midnight.exe	midday.exe
输入文件名	gloam.in	morning.in	midnight.in	midday.in
输出文件名	gloam.out	morning.out	midnight.out	midday.out
每个测试点时限	1.0 秒	1.0 秒	0.5 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	gloam.cpp	morning.cpp	midnight.cpp	midday.cpp
-----------	-----------	-------------	--------------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
-----------	--------------------------------------

## 收尾人 (gloam)

### 【题目背景】

被称作“图书馆”的光之高塔拔地而起，那片承载着生与死的秘境自然会吸引“收尾人”的驻足。

“收尾人”，书籍猎手的前身，终将葬身于“惨白管理者”统治下的“图书馆”中。

### 【题目描述】

作为一名事务所的代表，你接手管理着一个有很多收尾人的事务所。

由于管理混乱，存在着大量重复，冗余的表单，你需要将这些表单整理完毕，去除冗余的内容，然后整合形成一个新的表单。

现在存在着  $n$  张表单，每张表单上记录着  $A_i$  条信息。每条信息存在这样的三条内容：人名，编号和专精。

这些信息里面人名可能会重复，专精大概率会出现重复。但是保证每个人的编号不会出现重复。

整理后的表单应该包含这些表单的全部内容，同时重复的信息应该只会被记录一次。

### 【输入格式】

从文件 *gloam.in* 中读入数据。

第一行包含一个正整数  $n$ 。

接下来一共  $n$  张表单，对于第  $i$  张表单，第一行为一个正整数  $A_i$

接下来一共  $A_i$  行，每行包含一条信息的三个字符串，分别是人名，编号，专精。

### 【输出格式】

输出到文件 *gloam.out* 中。

输出新的表单，输出顺序按照专精，编号，人名的顺序进行比较（也就是专精不同，按照专精字典序顺序输出，编号，人名同理）。每条信息按照输入格式输出。

### 【样例 1 输入】

```
1 4
2 2
3 Roland BOOK1 seven
```

```
4 Angelica BOOK2 seven
5 3
6 mouse mouse mouse
7 Shao shao six
8 Rowell Rowell six
9 3
10 mouse mouse mouse
11 Roland BOOK1 seven
12 Angelica BOOK2 seven
13 3
14 Mali WARP97 none
15 Tomi WARP21 none
16 Tomali whatsthat battle
```

**【样例 1 输出】**

```
1 Tomali whatsthat battle
2 mouse mouse mouse
3 Tomi WARP21 none
4 Mali WARP97 none
5 Roland BOOK1 seven
6 Angelica BOOK2 seven
7 Rowell Rowell six
8 Shao shao six
```

**【样例 2 输入】**

```
1 2
2 15
3 teto teto teto
4 miku miku miku
5 yukari yukari yukari
6 akari akari akari
7 teto teto teto
8 rimi rimi rimi
9 yukari yukari yukari
10 akari akari akari
```

```
11 gumi gumi gumi
12 yukari yukari yukari
13 akari akari akari
14 teto teto teto
15 luka luka luka
16 moca moca moca
17 gumi gumi gumi
18 10
19 aoi aoi aoi
20 akane akane akane
21 rimi rimi rimi
22 yukari yukari yukari
23 akari akari akari
24 gumi gumi gumi
25 yukari yukari yukari
26 akari akari akari
27 teto teto teto
28 luka luka luka
```

### 【样例 2 输出】

```
1 akane akane akane
2 akari akari akari
3 aoi aoi aoi
4 gumi gumi gumi
5 luka luka luka
6 miku miku miku
7 moca moca moca
8 rimi rimi rimi
9 teto teto teto
10 yukari yukari yukari
```

### 【样例 3】

见选手目录下的 *gloam/gloam3.in* 与 *gloam/gloam3.ans*。

**【子任务】**

对于测试点 1, 2, 3, 满足  $n = 1, A_i \leq 50$

对于测试点 4, 5, 6, 保证所有人名, 编号, 专精之间没有重复

对于测试点 7, 8, 9, 10, 无特殊性质

保证全部信息数量小于  $5 \times 10^3$ , 每个字符串大小不会超过 10 且只有字母和数字

## 委托 (morning)

### 【题目背景】

从烦琐的跑腿，勘察，到雇凶杀人。只要付得起钱，他们就会接下任何工作。

他们受托于“事务所”，“辛迪加”，甚至是“翼”。从微乎其微的琐事，到惨绝人寰的要事。他们，无所不能。

### 【题目描述】

作为一名强大的收尾人，正如前文所说，你接下了来自  $L$  公司的  $n$  项委托。

每个委托的难度是不一样的，不过，你很强大，完全不用担心完成不了委托。只是不同的委托，因为难度不同，完成的时间也是不同的，对于委托  $i$  来说，你需要花费  $A_i$  的时间来完成。

这些委托之间互相都存在相互的联系，当你完成了其中任意一个委托时，其他所有的委托都因此变得更加简单。具体来说，当你完成了第  $i$  件委托时，会使得其他所有委托的完成需要的时间减少  $i$ （当然，完成时间不可能变成负数，如果有的委托的完成需要的时间变成小于等于 0 时，这个委托你也立刻完成）。

你只有完成了一个委托，才可以去完成其他委托。

你听说  $L$  公司最近出现了一些奇怪的东西，为了安全起见，你需要尽快完成这些任务，然后赶紧离开。

那么，你至少需要多少时间才能完成全部的任务呢？

### 【输入格式】

从文件 *morning.in* 中读入数据。

第一行输入一个正整数  $n$

接下来一行一共  $n$  个正整数，第  $i$  个为  $A_i$

### 【输出格式】

输出到文件 *morning.out* 中。

输出最少的完成时间。

### 【样例 1 输入】

```
1 5
2 6 6 6 5 5
```

**【样例 1 输出】**

1 18

按照顺序完成委托即可

每个任务需要的时间为:  $6 + 6 - 1 + 6 - 2 + 5 - 3 + 5 - 4 = 18$

**【样例 1 解释】****【样例 2】**

见选手目录下的 *morning/morning2.in* 与 *morning/morning2.ans*。

该样例满足特殊性质

**【样例 2 解释】****【样例 3】**

见选手目录下的 *morning/morning3.in* 与 *morning/morning3.ans*。

该样例满足 100% 的数据

**【子任务】**

对于 50% 的数据,  $N \leq 10$

另有 20% 的数据, 对于  $i \leq n$ ,  $A_i \leq A_{i+1}$

对于 100% 的数据,  $N \leq 10^5, A_i \leq 10^6$

## 爪牙 (midnight)

### 【题目背景】

将世界的奥秘玩弄于股掌之间，乃是“首脑”，“眼线”与“爪牙”至高无上的特权。

他们无人能挡，“巢”与他们的故事永不落幕。

### 【题目描述】

作为一名 L 公司的员工，正如前文所说，“爪牙”入侵了你们公司，你的同事们现在正在迎战。

不过他们的战斗，其实与你没有什么关系，你没有很强的战斗力，一旦被牵连了，可能轻则受伤，重则死亡。

因此，你最重要的任务，便是逃离这里，免得被他们的战斗所波及。

通过监控等方法，你现在获得了一些信息——L 公司可以认定是一个  $n \times m$  的空地（当然，这里忽略了公司里面的各种设施），地图上存在一些交战点（用 - 表示），如果经过了交战点，会被他们的战斗波及一次。

你每步可以走到上下左右的其中一格。

有些时候，你无论怎么走，都会被波及。

因此你需要找到要从当前位置（S）出发，到达唯一的出口（E），使得受到的波及次数最少的路径，考虑到输出答案可能会过大，你只需要输出被波及了几次即可。

### 【输入格式】

从文件 *midnight.in* 中读入数据。

请注意，由于地图很大，输入地图请以 `getchar()` 读入

第一行输入两个正整数，分别是地图的长和宽  $n, m$

接下来一共  $n$  行，每行包含  $m$  个字符。表示这张地图

输入字符只会包含以下四种：

S：起点

E：终点，起点和终点最多只有 1 个

-：交战点

.: 空地，没有特殊情况。

### 【输出格式】

输出到文件 *midnight.out* 中。

从起点到终点最少会被波及几次。



**【样例 1 输入】**

```
1 5 6
2 E.....
3 -----
4 ---...
5 ...---
6 .....S
```

**【样例 1 输出】**

```
1 2
```

**【样例 1 解释】**

一种方法：一路向上走到左上角，然后一路向左走到终点。

**【样例 2】**

见选手目录下的 *midnight/midnight2.in* 与 *midnight/midnight2.ans*。

**【子任务】**

对于 45% 的数据， $n, m \leq 4$

对于 80% 的数据， $n, m \leq 1000$

另有 10% 的数据， $n = 1, m \leq 3000$

对于 100% 的数据， $n \leq 3000, m \leq 3000$

## 武器 (midday)

### 【题目背景】

他们无休无止地搜罗着精良的武器。无论是“翼”的科技，还是“后巷”的发明，亦或是“郊区”的战利品，乃至“废墟”的遗物...

他们的所作所为，无非是紧握手中的武器，一如既往地用暴力解决眼中的一切。

### 【题目描述】

作为一名收尾人，正如前文所说，你需要高级的武器来强化自己，无论这个武器来自哪里，出自何方，只要可以强化自己的战斗力，就是优秀的武器。

不同的武器可以通过各种方法提供战斗力，包括但不限于身体强化，力量提升。当然，这些东西太具体了，我们就统统认为，对于第  $i$  把武器，其可以提供  $B_i$  的战斗力，装备上时，你的战斗力会增加  $B_i$ 。

当然，这些强大的武器使用会有各种限制，其会存在两条限制：

- 装备武器也需要一定的战斗力，对于第  $i$  把武器，你自身的战斗力至少需要  $T_i$  才可以拿起。
- 除了拿起武器的时刻外，你无法同时携带两把武器，每当你携带武器的数量大于等于 2 把时，你需要舍弃至只剩下一把武器。被舍弃的武器所提供的战斗力，自然就丢失了。

很明显，打铁还需自身硬，你自身需要有强大的战斗力，才可以装上这些武器。

那么，问题很简单，给定  $n$  把武器和  $q$  次询问，每次询问给定你的初始战斗力  $X$ ，问你最大可以取得的战斗力是多少？

### 【输入格式】

从文件 *midday.in* 中读入数据。

第一行包含两个正整数，分别是  $n, q$

第二行包含  $n$  个正整数，第  $i$  个数为  $B_i$

第三行包含  $n$  个正整数，第  $i$  个数为  $T_i$

第四行包含  $q$  个正整数，第  $i$  个数为第  $i$  次查询的  $X$

### 【输出格式】

输出到文件 *midday.out* 中。

输出  $q$  行，每行包含一个正整数，为第  $i$  次询问最大可以取得的战斗力。

**【样例 1 输入】**

```
1 3 2
2 21 15 7
3 4 6 12
4 6
5 1
```

**【样例 1 输出】**

```
1 27
2 1
```

**【样例 1 解释】**

对于第一组查询，可以拿起武器 1,2，显然选择拿武器 1 更优秀。拿起武器后不需要拿其他武器。

**【样例 2 输入】**

```
1 5 10
2 1 2 3 4 5
3 2 4 6 8 10
4 1
5 2
6 3
7 4
8 5
9 6
10 7
11 8
12 9
13 10
```

**【样例 2 输出】**

```
1 1
2 3
3 5
4 7
5 9
6 11
7 12
8 13
9 14
10 15
```

### 【样例 2 解释】

对于第六组查询，可以拿起武器 3，此时战斗力到达 9，可以拿起武器 4，拿起武器 4 放下武器 3 后战斗力到达 10，可以拿起武器 5，然后放下武器 4。

最终战斗力为  $5 + 6 = 11$ 。

### 【样例 3】

见选手目录下的 *midday/midday3.in* 与 *midday/midday3.ans*。

### 【子任务】

对于 40% 的数据， $n, q \leq 1000$

对于 65% 的数据， $n \leq 10^3, q \leq 10^5; X, T_i, B_i \leq 10^5$

另有 15% 的数据，保证所有的  $X$  单调递增

对于 100% 的数据， $n, q \leq 10^5; X, T_i, B_i \leq 10^9$