

清华大学计算机系程序设计水平测试

推研机试

A 卷

时间：2018 年 9 月 16 日 13:30 ~ 17:30

题目名称	贪吃蛇	过去的项链	八四的鹅卵石
题目类型	传统型	传统型	传统型
输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB
测试点/包数目	6	20	10
测试点是否等分	否	是	是

注意事项

请大家仔细阅读题，建议从简单到困难完成各题子任务，避免“陷入某道题，忽略其他题”情况发生。

贪吃蛇 (snake)

【题目描述】

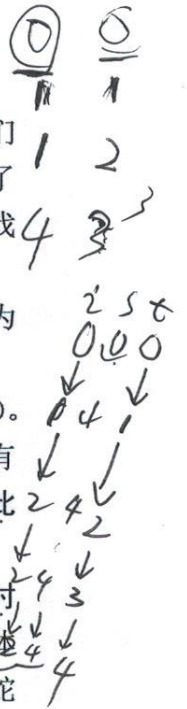
Bakser 和 Wangxz 以及一些朋友们想要玩贪吃蛇对战游戏。作为硬核选手，他们合作编写了一份贪吃蛇程序以进行游戏。不幸的是，他们对于某一局游戏的结果产生了争执，都认为对方编写的逻辑存在 bug。他们将游戏的 log 导出，并请你编写一份游戏逻辑以判断游戏的结果，游戏的具体规则如下：

- 游戏地图为 64×64 的网格地图，左上角格子坐标为 $(0,0)$ ，右下角格子坐标为 $(63,63)$ 。格子 (x,y) 代表位于从上往下数第 x 行，从左往右数第 y 列的格子。
- 游戏可以有任意多个玩家，每个玩家有一定数量的蛇（数量在游戏开始时设定）。同一时刻每个玩家只能有最多一条蛇在场上。当某个玩家的蛇死亡后，若他还有剩余的蛇，则在一段时间后按照系统指令在出现一条新的蛇供该玩家控制。若此玩家的蛇死亡且没有剩余的蛇，则他游戏失败。
- 每条蛇出生时长度均为 2，在系统指定位置出生，保证在有效出生指令执行时刻该指定位置一定为空，若出生指令涉及的坐标在指令执行的时刻被其他物体（蛇、墙、豆子）占据或坐标不合法，该出生指令无效，保证出生指令指定的蛇的头尾位置一定相邻。蛇出生后若无玩家操作则静止不动，直至玩家发出移动指令后才开始移动（若玩家指定的移动方向恰好从头指向尾，视为头尾交换），并且直至死亡再也不会停止。
- 蛇每一秒移动一格，若玩家在这一秒指定了蛇头的移动方向，则蛇头按指定方向移动；若未指定，则按上一秒的方向移动；若指定的移动方向与原先移动方向正好相反，则不会生效。若在某一秒蛇头移动到了豆子的位置，则豆子被吃掉，同时蛇身长度增加。每吃掉 1 个豆子，蛇身长度增加 1。
- 蛇头碰撞到自己的蛇身、对方蛇的蛇身、墙或越过地图边界，即会死亡。若多条蛇的蛇头发生碰撞（无论正撞或侧撞），长度最长者存活，其他蛇死亡；若碰撞时刻恰好有多条蛇且长度均为最大，则发生碰撞的所有蛇均死亡。
- 系统会不定期通过指令指示随机撒一些豆子在某些格子上，或者在一些格子上随机生成墙。若该指令涉及的某些坐标在指令执行的时刻被其他物体（蛇、墙、豆子）占据或坐标不合法，整条指令作废；若该指令中有重复的坐标，只在该坐标生成一个豆子（墙）。

你将得到按时间顺序给出的玩家操作记录（若某玩家在同一秒进行了多个移动操作，只取记录顺序最后的一个，无论它是否有效）和系统生成墙和豆子的记录，请你根据这份记录判断哪位玩家取得了胜利。

若同一秒内同时有系统的操作和玩家的操作，系统的操作会首先按记录顺序执行，玩家的操作按照记录的先后顺序依次执行。

在同一秒内，每条蛇的移动操作也有先后顺序，若这一秒内玩家通过有效的操作指定了蛇的移动方向，则这条蛇在这条操作得到执行时发生移动，否则按原有顺序移动；



按原有方向移动的蛇，在这一秒内所有操作执行完成后，按最开始给出玩家 ID 时的顺序发生移动。发生移动后触发的事件（如蛇变长、碰撞等）均在当前移动发生后立即结算。

若某条指令执行时刻条件不能满足或与上述规则冲突，如玩家发出移动指令但场上并没有他的蛇存在，或系统给出了蛇出生指令但该玩家已经没有剩余的蛇、系统给出了蛇出生指令但该玩家场上尚有存活的蛇等，则该指令自动忽略。

在记录内容结束时场上仍有存活或剩余的蛇的玩家（们）取得游戏的胜利，若记录结束时所有玩家的蛇均死亡且没有剩余的蛇，则没有玩家取得胜利。

【输入格式】

从标准输入读入数据。输入文件的第一行包含一个整数 n ，表示共有 n 个玩家。

接下来 n 行，每行一个字符串（保证仅由英文字母、下划线、阿拉伯数字组成，不会超过 20 个字符，且不会是 System、Bean 或 Wall）和一个正整数，字符串表示一个玩家的 ID，正整数表示该玩家的蛇的数量。

接下来一行一个整数 L ，表示共有 L 条记录。

接下来有 L 行，每一行表示一条记录，行内的字符串或整数以一个空格分开。

每一行的开头均为一个非负整数 S ，表示这条操作发生在游戏开始后的第几秒。

若接下来的字符串为某个玩家的 ID，则表示是这个玩家进行的操作。再接下来的字符串若为 UP、DOWN、LEFT、RIGHT，则分别表示操作自己的蛇向上、下、左、右移动。

若 S 后面的字符串为 System，表示是游戏系统进行的操作，若接下来的字符串为 Wall，表示生成墙，若接下来字符串为 Bean，表示生成豆子，接下来跟着一个正整数 n ，后面跟着 $2n$ 个整数 $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ ，代表在坐标分别为 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 的格子添加墙或一个豆子。若 System 后面的字符串是一个玩家的 ID，表示该玩家的蛇出生了，后面跟着四个整数 x_1, y_1, x_2, y_2 ，表示该蛇蛇头在 (x_1, y_1) ，蛇身在 (x_2, y_2) 。

保证所有 S 不超过 3,000，保证后给出的 S 一定不小于先给出的 S 。

保证输入数据中所有整数均在 32 位带符号整型范围内。

【输出格式】

输出到标准输出。

输出 n 行。

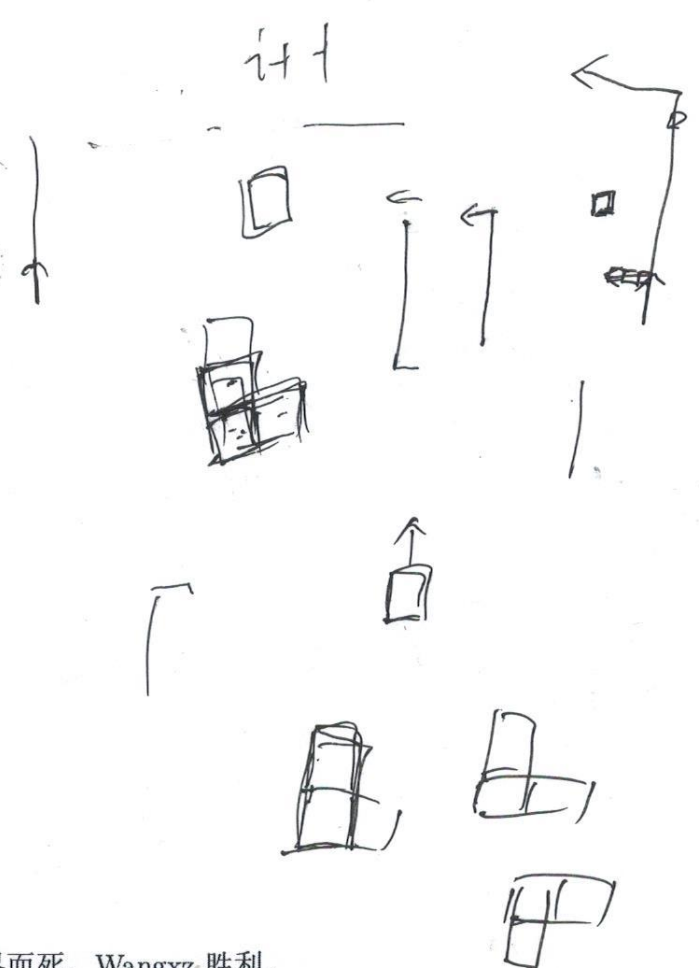
每行开头一个字符串，为一个玩家的 ID，若该玩家胜利，后面跟着一个字符串 WIN，若该玩家失败，后面跟着一个整数表示该玩家在第多少秒失败。

各玩家按游戏失败的时间顺序输出，若同一时间多个玩家失败，按输入最开始给出玩家的相对顺序输出，胜者最后输出，若有多个胜者，按照输入最开始给出玩家的相对顺序输出所有胜者。

【样例输入】

```

2
Wangxz 1
Bakser 1
8
0 System Wangxz 1 0 0 0
0 System Bakser 62 63 63 63
1 Bakser UP
1 Wangxz DOWN
5 System Wall 2 20 20 26 23
30 Wangxz RIGHT
38 Wangxz DOWN
65 System Wall 2 20 20 26 23
    
```



【样例输出】

```

Bakser 63
Wangxz WIN
    
```

【样例解释】

第 63 秒, Bakser 的蛇越过地图上边界而死, Wangxz 胜利。

【子任务】

子任务	n	L	每个玩家蛇的数量	特殊性质	分值
1	$= 1$	$\leq 10^2$	≤ 1	没有墙和豆子	15
2	$= 1$	$\leq 10^2$	≤ 1	没有豆子	15
3	$= 1$	$\leq 10^2$	≤ 1	无	15
4	$= 2$	$\leq 3,000$	≤ 1	无	15
5	$= 10$	$\leq 3,000$	≤ 1	无	20
6	$= 10$	$\leq 3,000$	≤ 10	无	20

过去的项链 (reverse)

【题目描述】

过去有一个项链。这个项链上串了 N 个珠子，其中第 i 个珠子上有 a_i 的能量值 (a_i 不一定非负)。

对于一段连续的珠子，过去定义其美丽程度为它们的能量值之和。自然地，一个项链的美观值定义为它的所有连续子段的美丽程度的最大值。

过去得到了一把神奇剪刀，他可以把这个项链的恰好一个连续子段剪下来，然后翻转这个子段，再拼接回去得到一个新项链。

过去想利用这把剪刀，得到一个美观值尽可能大的新项链。那么问题来了：请你告诉过去，他的新项链的美观值最大可以是多少，以及为了达到这个最大值，我们应该选择原项链的哪一段去切它。

【输入格式】

从标准输入读入数据。

输入第一行包含一个正整数 N ，保证 $1 \leq N \leq 2 \times 10^5$ 。

接下来一行 N 个整数，描述序列 $\{a_i\}$ ，这里下标从 1 开始，保证 a_i 的绝对值不超过 10^8 。

【输出格式】

输出到标准输出。

输出包含两行。

第一行包含一个整数，表示美观值的最大值。第二行包含两个整数 l, r ，表示你的方案中选择把项链的 a_l, a_{l+1}, \dots, a_r 这一段切下来、翻转、拼接回去。

注意：这里输出的是一个闭区间。特别地，当 $l > r$ 时，表示选择了项链的 $a_l, a_{l+1}, \dots, a_N, a_1, a_2, \dots, a_r$ 这一段；当 $l = r$ 时，单个珠子被翻转相当于什么也没有发生。

【样例 1 输入】

```
6
3 2 -1 1 -1 5
```

【样例 1 输出】

```
11
3 4
```

【样例 1 解释】

翻转 a_3, a_4 之后, 美丽程度的最大值在子段 $[5, 3, 2, 1]$ 上取到, 故答案为 11。

【样例 2】

见题目目录下的 *2.in* 与 *2.ans*。

【子任务】

- 子任务 1 (30 分): $N \leq 50$
- 子任务 2 (25 分): $N \leq 400$
- 子任务 3 (25 分): $N \leq 3000$
- 子任务 4 (20 分): $N \leq 2 \times 10^5$

八四的鹅卵石 (lics)

【题目描述】

小八和小四去海边度假，海边有一片美丽的鹅卵石滩。小八在鹅卵石滩上捡了 n 块美丽的鹅卵石，并把它们排成一个序列，其中排第 i 位的鹅卵石的美丽度为 a_i 。小八想从里面按照原序列的顺序挑选出一个鹅卵石的子序列，使得在这个子序列里的最后一块鹅卵石的美丽度不比前一块低。小八还想知道，他这么做能得到的最长的子序列长度是多少。

小四认为这个问题很 naive，于是他决定将小八的鹅卵石序列首尾相连，组成一个鹅卵石环，然后计算在这个环上的满足要求的最长子序列的长度。

形式化地，对于一个子序列 $a_{x_1}, a_{x_2}, \dots, a_{x_t}, \dots, a_{x_m}$ ($1 \leq x_1 < x_2 < \dots < x_m \leq n$)，它是一个满足要求的子序列当且仅当它满足以下两个条件：

1. $a_{x_1} \leq a_{x_2} \leq \dots \leq a_{x_m}$ 。
2. $1 \leq x_1 < x_2 < \dots < x_m \leq n$ 或 $1 < x_1 < x_2 < \dots < x_t \leq n, 1 \leq x_{t+1} < x_{t+2} < \dots < x_m < x_1$ ($\exists t \in \mathbb{N}, 1 < t < m$)。

【输入格式】

从标准输入读入数据。

输入第一行包含一个整数 T ，代表数据组数。

接下来有 $2T$ 行，每两行代表一组数据。

每组数据的第一行包含一个整数 n ，代表鹅卵石的个数。

第二行包含 n 个非负整数，依次是 a_1, a_2, \dots, a_n ，代表 n 个鹅卵石各自的美丽程度。

保证所有数据随机生成。具体地，每一项 a_i 独立地从 $[1, n]$ 内的整数中等概率选取。

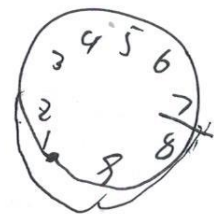
【输出格式】

输出到标准输出。

输出共 T 行，每行一个正整数 m ，表示该组数据中最长的满足要求的子序列的长度。

【样例输入】

```
2
3
3 1 2
10
1 3 8 8 1 7 9 3 10 10
```



8 9 1 2 7 4 5 6 7

【样例输出】

3

7

【样例解释】

对第一组数据，选取的子序列为 a_2, a_3, a_1 。

【子任务】

测试点	n	T
1,2	$= 10$	≤ 10
3,4	$= 10^2$	
5,6	$= 10^3$	
7,8,9,10	$= 10^4$	