

2023 年信息学国庆集训班 普及组

第 3 天

时间：2023 年 10 月 3 日 8:00 ~ 11:30

题目名称	选举	移动	魔力球	圆形广场
题目类型	传统型	传统型	传统型	传统型
输入文件名	election.in	move.in	magicball.in	circle.in
输出文件名	election.out	move.out	magicball.out	circle.out
每个测试点时限	1.0 秒	1.0 秒	1.0	2.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB

提交源程序文件名

对于 C++ 语言	election.cpp	move.cpp	magicball.cpp	circle.cpp
-----------	--------------	----------	---------------	------------

注意事项与提醒（请选手务必仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件时需要 建立子文件夹。
4. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
5. 程序可使用的栈内存空间限制与题目的内存限制一致。
6. 题目难度大致按出题人预期排序，不代表题目按每个人的难度排序。

选举 (election)

【题目描述】

小 Z 的学生会正在举行换届选举！现在有 n 个人在竞选 m 个职务。其中第 i 个人的学号为 i ，且正在竞选第 a_i 个职务，获得的票数为 b_i 。每个职务获得票数最多的人当选，若获得票数相同则学号较小的人当选。小 Z 想知道最终当选每个职务的人的学号是多少。

数据保证每个职务均有至少一人参与竞选。

【输入格式】

从文件 *election.in* 中读入数据。

第一行包含两个用空格分隔的正整数 n, m ，分别表示竞选人数和职位数。

接下来 n 行，每行两个正整数 a_i, b_i 表示学号为 i 的人参与竞选的职务和已获得的票数。

【输出格式】

输出到文件 *election.out* 中。

输出 m 个用空格分隔的整数，表示最终当选第 i 个职务的人的学号。

【样例输入 1】

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

【样例输出 1】

```
2 1
```

【样例输入 2】

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

【样例输出 2】

```
5 1 3 2 4
```

【数据范围】

对于 20% 的数据，保证 $1 \leq n \leq 20$;

对于另外 20% 的数据，保证 $n = m$;

对于 100% 的数据，保证 $1 \leq m \leq n \leq 10^5, 1 \leq b_i \leq n, 1 \leq m \leq n$ 。

移动 (move)

【题目描述】

小 Z 正在玩一款游戏！这款游戏在一个 2D 的网格图上进行，初始主角在原点 $(0,0)$ 。小 Z 会输入一段操作序列，仅包含字符“RULD”，每个字符代表以下操作之一：

- ‘R’：小 Z 从 (x,y) 移动到 $(x+1,y)$ ；
- ‘L’：小 Z 从 (x,y) 移动到 $(x-1,y)$ ；
- ‘D’：小 Z 从 (x,y) 移动到 $(x,y-1)$ ；
- ‘U’：小 Z 从 (x,y) 移动到 $(x,y+1)$ ；

但由于地图上存在障碍，有可能某些操作中移动到的格子上存在着障碍。当撞上障碍时，这一步移动将会取消，并且将会留在原地。

小 Z 不知道地图的信息，但知道地图保证原点 $(0,0)$ 上没有障碍。现在，小 Z 给你了上述的操作序列，小 Z 想知道按顺序执行完所有操作后，所有可能的终点位置？

【输入格式】

从文件 *move.in* 中读入数据。

第一行包含一个正整数 n ；

第二行包含一个长度为 n 且只包含字符 *RULD* 的字符串，表示小 Z 的操作序列。

【输出格式】

第一行输出一个正整数 k 表示可能的终点位置；

接下来 k 行每行包含两个整数 x,y ，表示一个可能的终点位置。

注意当 $k \geq 2$ 时，需保证优先按 x 升序输出， x 相同时按 y 升序输出。

【样例输入 1】

```
2
RU
```

【样例输出 1】

```
4
0 0
0 1
1 0
1 1
```

【样例输入 2】

3
RDL

【样例输出 2】

5
-1 -1
-1 0
0 -1
0 0
1 -1

【样例输入 3】

见选手目录下的 *move/move3.in*

【样例输出 3】

见选手目录下的 *move/move3.out*

【数据范围】

对于 20% 的测试点，保证 $n \leq 3$ ；

对于 60% 的测试点，保证 $n \leq 15$ ；

对于另外 20% 的测试点，保证操作序列包含 *RL* 或者 *UD*；

对于 100% 的测试点，保证 $n \leq 20$ 。

魔力球 (magicball)

【题目描述】

小鸾无意之中获得了一个神奇的魔法棒，魔法棒会源源不断地吸取新的能量来增强魔力，并且可以从顶端释放魔力来施放魔法。

魔法棒中储存魔力靠的是许许多多的魔力球。初始时（第 0 个周期）魔法棒只在靠近顶端的位置有一个大魔力球。每一个魔法周期，魔法棒都会吸取新的能量来增强魔力，而魔法棒增强魔力的本质是其中所有的魔力球从外界吸取能量。魔力球吸取能量后会发生以下的变化：

- 小魔力球：小魔力球吸取能量后会变成大魔力球；
- 大魔力球：大魔力球吸取能量后会在大魔力球靠近顶端的一边新生成 2 个小魔力球（紧挨着大魔力球）。

现在恰好过了 n 个魔法周期，小鸾需要使用魔法棒中的魔力来施放魔法。由于魔法棒只能从顶端释放魔力，而魔法棒中的魔力球紧凑地排成一列，如果靠近顶端的魔力球不消耗则无法使用后面的魔力球。只有大魔力球可以为施放魔法提供魔力，小魔力球在施放魔法时无法提供魔力只能白白地被消耗掉。

小鸾在此次施放前没有施放过任何魔法，并且此次施放需要用到 k 个大魔力球。小鸾想知道施放这次魔法至少需要消耗多少个魔力球（包括大魔力球和小魔力球）。

【输入格式】

从文件 *magicball.in* 中读入数据。

输入一行两个整数 n, k ，分别表示魔法周期和施放魔法所需大魔力球。

【输出格式】

输出到文件 *magicball.out* 中。

输出一行一个整数，表示需要消耗的魔力球个数。

【样例输入 1】

3 3

【样例输出 1】

7

【样例解释 1】

如果用大写字母 A 表示大魔力球，小写字母 a 表示小魔力球。

则在第一个样例中，3 个魔法周期后魔法棒的魔力球从顶端开始依次为 aaAaaAAaaA，则释放 3 个大魔力球至少需要 7 个魔力球。

【样例输入 2】

4 5

【样例输出 2】

7

【样例输入 3】

25 100000

【样例输出 3】

200002

【数据范围】

对于 20% 的数据，保证 $1 \leq n \leq 5$ 。

对于 40% 的数据，保证 $1 \leq n \leq 25$ 。

对于另外 10% 的数据，保证 $k = 1$ 。

对于 100% 的数据，保证 $1 \leq n \leq 50$ ， k 大于等于 1 且保证不超过恰好 n 个魔法周期后总共有大魔力球个数。

圆形广场 (circle)

【题目描述】

小鸾决定在一个巨大的圆形广场上，开展一场大型宣讲活动。

为了保证活动秩序，小鸾邀请了 n 位同学参加宣讲活动，并将他们安排在圆形广场的圆周上。

小鸾设计了一种活动队形，在她的设计中，所有的同学都坐在圆形广场的边缘，第 i 位同学与第 $i + 1$ 位同学的就座间隔为 a_i ， a_i 描述的是两位同学之间的圆弧长度。特殊地， a_n 描述的是第 n 位同学与第 1 位同学的间隔距离。

小鸾对于活动的有序和整齐程度有着过分的追求，为了衡量一种队形是否有序，她定义了队形的有序度。一个队形的有序度是指同学们组成的中心对称图形的个数。具体的，有序度是序列 $1 \leq p_1 < p_2 < \dots < p_l \leq n$ 的数量，使得第 p_1, p_2, \dots, p_l 位同学依次连边成环，组成的多边形是中心对称图形。

小鸾想要知道，她设计的队形的有序度是多少。她知道这个数可能很大，因此你只需要告诉她这个值对 998244353 取模的结果。

【输入格式】

从文件 `circle.in` 中读入数据。

共 $n + 1$ 行，第一行为正整数 n ，表示同学的个数。

接下来 n 行，第 i 行为 a_i ，表示同学之间的间隔距离，其具体定义见题目描述。

【输出格式】

输出到文件 `circle.out` 中。

输出一个数，为小鸾设计的队形的有序度对 998244353 取模的值。

【样例输入 1】

```
8
1 2 2 3 1 1 3 3
```

【样例输出 1】

```
4
```

【样例解释 1】

由序列 $\{1, 2, 5, 6\}$ 、 $\{2, 4, 6, 7\}$ 、 $\{1, 4, 5, 8\}$ 和 $\{1, 2, 4, 5, 6, 8\}$ 对应的同学组成的多边形均中心对称。

【样例输入 2】

见选手目录下的 `circle/circle2.in`

【样例输出 2】

见选手目录下的 *circle/circle2.out*

【样例输入 3】

见选手目录下的 *circle/circle3.in*

【样例输出 3】

见选手目录下的 *circle/circle3.out*

【数据范围】

对于 30% 的数据, $2 \leq n \leq 10, 1 \leq a_i \leq 100$;

对于 50% 的数据, $2 \leq n \leq 10^3, 1 \leq a_i \leq 10^6$;

对于 80% 的数据, $2 \leq n \leq 10^6, 1 \leq a_i \leq 10^9$;

对于另外 10% 的数据, 满足所有 a_i 的和为奇数;

对于 100% 的数据, $2 \leq n \leq 5 \times 10^6, 1 \leq a_i \leq 10^{10}$ 。