

正睿青少年信息学奥林匹克竞赛

2025 暑假 AB 班联考

Day2

时间：2025 年 8 月 7 日 8:00 ~ 12:00

题目名称	望远镜	追逐	套娃
题目类型	传统题	传统题	传统题
目录	telescope	chase	doll
可执行文件名	telescope	chase	doll
输入文件名	telescope.in	chase.in	doll.in
输出文件名	telescope.out	chase.out	doll.out
每个测试点时限	1.5 秒	1.0 秒	0.5 秒
内存限制	1024 MB	1024 MB	1024 MB
测试点数目	25	25	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	telescope.cpp	chase.cpp	doll.cpp
-----------	---------------	-----------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。赛后正式测试时将以选手留在题目目录下的源代码为准。
2. `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. 因违反以上两点而出现的错误或问题，申诉时一律不予受理。
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. 选手提交的程序源文件必须不大于 100KB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。
8. 选手可在比赛主页绑定工作目录后点击【保存预提交】将工作目录下所有代码一次性提交到评测系统。

望远镜 (telescope)

【题目描述】

小 Z 有一台望远镜。不同于传统的望远镜，这台望远镜有 m 种放大镜片，第 i 种放大镜片能将物体放大成 i 倍，且有无限个。

小 Z 极具冒险精神，他想通过这台自制望远镜观测 clb 星球。不过由于 clb 星球对于普通人类来讲过于遥远，因此对放大倍数要求极高。小 Z 想通过对于放大镜片的组合来观测，他总共选取了 n 个放大镜片，第 i 个放大镜片为正整数 a_i ，则总放大倍数为 $\prod_{i=1}^n a_i$ ，第 i 个放大镜片可以在 $[1, m]$ 之间任意选择。

出于对 clb 星球的敬畏，小 Z 不希望将倍数放得太大，因此他希望总放大倍数不超过 m ，记 $\prod_{i=1}^n a_i \leq m$ 。

由于观测 clb 星球的方法非常多，因此只要求出对 $10^9 + 7$ 取模后的值。

【输入格式】

一行两个正整数 n, m 表示放大镜片数量和最大放大倍率。

【输出格式】

一行一个非负整数 ans 表示答案。

【样例 1 输入】

1 5 2

【样例 1 输出】

1 6

【样例 1 解释】

6 种方案分别为：

1. $\{1, 1, 1, 1, 1\}$;
2. $\{2, 1, 1, 1, 1\}$;
3. $\{1, 2, 1, 1, 1\}$;
4. $\{1, 1, 2, 1, 1\}$;
5. $\{1, 1, 1, 2, 1\}$;
6. $\{1, 1, 1, 1, 2\}$ 。

【样例 2 输入】

1

16 18

【样例 2 输出】

1

10453

【样例 3 输入】

1

292 286

【样例 3 输出】

1

271545773

【样例 4 输入】

1

4997 4946

【样例 4 输出】

1

804741638

【样例 5 输入】

1

49997653 49999597

【样例 5 输出】

1

265626389

【数据范围与提示】

对于所有测试数据，保证：

- $1 \leq n, m \leq 5 \times 10^7$ 。

测试点编号	$n \leq$	$m \leq$
1, 2	5	5
3, 4	20	20
5 ~ 7	50	50
8, 9	300	300
10 ~ 12	1000	1000
13, 14	5000	5000
15 ~ 17	5000	5×10^7
18 ~ 20	5×10^7	5000
21, 22	10^5	10^5
23	10^6	10^6
24	10^7	10^7
25	5×10^7	5×10^7

追逐 (chase)

【题目描述】

Z 国风景秀丽，由 n 个城市组成，每个城市都别具一格，形成了丰富多彩的大国江山。由于 Z 国的贸易出现危机，财政紧张所以仅有 $n - 1$ 条双向道路连接着这 n 座城市。其中第 i 条双向道路连接了城市 u_i 和城市 v_i 。为了保证城市的方便快捷，这 $n - 1$ 条道路能保证一定能从一座城市到达另一座城市。

在五一假期，Alice、Bob 不约而同地选择出去自驾。他们各自的自驾旅行是一条从家到一座城市的最短路径。Alice 会先选择一个起点城市，Bob 则会根据 Alice 选择的的城市选择他的起点城市。

由于两人数年的恩怨，因此他们想在这个假期里一较高低——谁的旅行时间更长，如果时间相同则 Bob 获胜。此外，两人都有一颗好奇心，因此他们不想去他们自己或对方已经去过的城市。

具体来说，在假期里的每一天会依次发生如下两件事：

1. Alice 先选择一个和他当前所在城市有双向道路连接的城市，并且要求这座城市两人此前均未曾经过，如果无法找到这座城市，则 Alice 输掉这场竞争，Bob 获胜；
2. Bob 后选择一个和他当前所在城市有双向道路连接的城市，并且要求这座城市两人此前均未曾经过，如果无法找到这座城市，则 Bob 输掉这场竞争，Alice 获胜。

由于显然竞争会在有限轮数内结束，因此小 Z 想知道在两人都采取最优策略的情况下，谁会赢得这场竞争。

由于每年五一假期，两人都会进行一场竞争，所以请你对 T 次竞争都进行判断。

【输入格式】

第一行一个正整数 T 表示竞争数量，对于每组竞争：

第一行一个正整数 n 表示 Z 国城市数量。

接下来 $n - 1$ 行，第 i 行两个正整数 u_i, v_i 表示第 i 条道路连接的两座城市编号。

【输出格式】

T 行，每行一个字符串，若 Alice 获胜，输出 **Alice**，否则输出 **Bob**。

【样例 1 输入】

```
1 2
2 3
3 3 2
4 1 2
5 5
6 3 2
7 3 4
8 2 5
9 2 1
```

【样例 1 输出】

```
1 Alice
2 Bob
```

【样例 1 解释】

对于第 1 组测试数据，Alice 选择点 2，不难发现最后 Alice 会获胜。

对于第 2 组测试数据，下面演示一种 Bob 获胜的情况：

1. Alice 选择点 1；
2. Bob 选择第 2；
3. Alice 无法选择下一个点，Bob 获胜！

【样例 2 输入】

见选手目录下的 `chase/chase2.in` 与 `chase/chase2.ans`。

该样例满足测试点 1 ~ 3 的约束条件。

【样例 2 输出】**【样例 3 输入】**

见选手目录下的 `chase/chase3.in` 与 `chase/chase3.ans`。

该样例满足测试点 10 的约束条件。

【样例 3 输出】**【样例 4 输入】**

见选手目录下的 `chase/chase4.in` 与 `chase/chase4.ans`。
该样例满足测试点 20 ~ 24 的约束条件。

【样例 4 输出】

套娃 (doll)

【题目描述】

小 Z 是一位很菜的算法竞赛选手，由于要出集训队互测，所以他开始艰难地随机 idea。



图 1 123

在经历了漫长的失败无果后，小 Z 终于出了一道有一定难度的题目。而本题便是众多失败中的一个。

小 Z 喜欢组装套娃，并构成不同的结构。在拼装的过程中，小 Z 产生了自己独特的审美。他规定：一组合法嵌套形如一串套娃依此嵌套形成一个链式结构。

具体地，我们如下定义合法一组合法嵌套：

- 1. 单独一个套娃是一组合法嵌套；
- 2. 一个套娃内部恰好有一组合法嵌套是一组合法嵌套。

例如，这是一组合法嵌套的展开形式：



图 2 oqrtsp

这也是一组合法嵌套：



图 3 2bd9723e4d30b4188fd55a99c3f918e3
这不是一组合法嵌套：



图 4 QQ_1754474432188

小 Z 有 n 个套娃，由于套娃的承重有限，所以第 i 个套娃内至多总共只能有 a_i 个套娃。现在，小 Z 想将这些套娃收起来放到柜子里，所以他将这些套娃组成了一些合法嵌套组。由于柜子的空间有限，所以小 Z 希望形成尽量少的组。

因为前途是美好的，每个人都有光明的未来，所以小 Z 想对每个前缀都求出最少形成的合法嵌套组的组数。

【输入格式】

第一行一个正整数 n ，表示总套娃个数。

第二行 n 个非负整数 a_i 表示第 i 个套娃内的最多套娃个数。

【输出格式】

一行 n 个正整数，表示前 i 个套娃最少形成的合法嵌套组的组数。

【样例 1 输入】

```
1 5
2 0 0 1 2 1
```

【样例 1 输出】

```
1 1 2 2 2 2
```

【样例 1 解释】

- 对于前 1 个套娃，显然最少需要分成 1 组；
- 对于前 2 个套娃，由于两个套娃都不能套进任何套娃，所以最优方案为它们分别分为 2 组；
- 对于前 3 个套娃，一组最优方案为 $\{0\}_1, 0$ ；
- 对于前 4 个套娃，一组最优方案为 $\{\{0\}_1\}_2, 0$ ；
- 对于前 5 个套娃，一组最优方案为 $\{\{0\}_1\}_2, \{0\}_1$ ；

【样例 2 输入】

```
1 5
2 0 2 3 0 1
```

【样例 2 输出】

```
1 1 1 1 2 2
```

【样例 3 输入】

```
1 10
2 1 0 3 1 1 1 2 0 1 3
```

【样例 3 输出】

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

【样例 4 输入】

见选手目录下的 `doll/doll4.in` 与 `doll/doll4.ans`。
该样例满足测试点 4 ~ 6 的约束条件。

【样例 4 输出】**【样例 5 输入】**

见选手目录下的 `doll/doll5.in` 与 `doll/doll5.ans`。
该样例满足测试点 13 的约束条件。

【样例 5 输出】**【样例 6 输入】**

见选手目录下的 `doll/doll6.in` 与 `doll/doll6.ans`。
该样例满足测试点 25 的约束条件。

【样例 6 输出】**【数据范围与提示】**

对于所有测试数据，保证：

- $1 \leq n \leq 10^6$;
- $0 \leq a_i < n$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 3	10	无
4 ~ 6	20	无
7	50	无
8	100	无
9	300	无
10	500	无
11	1000	无
12	2000	无
13	5000	无
14	7000	无
15	10^4	无
16, 17	5×10^4	无
18	10^5	无
19	2×10^5	无
20	3×10^5	无
21	5×10^5	无
22	7×10^5	无
23	10^6	A
24	10^6	B
25	10^6	无

- 特殊性质 A: 保证 $\forall i \in [1, n-1], a_i \leq a_{i+1}$;
- 特殊性质 B: 保证 $\forall i \in [1, n-1], a_i \geq a_{i+1}$ 。