

a_little_cute Round 2

题目名称	第一题	不是第一题	第三题	不是第三题
题目类型	普通	普通	普通	普通
输入文件	triple.in	array.in	price.in	travel.in
输出文件	triple.out	array.out	price.out	travel.out
时空限制	1s 512MB	2s 1024MB	3s 512MB	1.5s 512MB
测试点数目	10	8	25	20
测试点是否等分	是	否	是	是

编译选项:

```
-std=c++14 -w1,-stack=123456789 -O2
```

- 1.保证所有题目时限大于 std 运行时间的 2 倍。
- 2.AK 后请不要大声喧哗。
- 3.请先通读题面。若发现原题，可以告诉出题人，但不要讨论。
- 4.有且仅有 B.不是第一题 的样例输出文件后缀名为 .ans 。

A. 第一题

小 C 喜欢三元组。他认为一个整数三元组 (x, y, z) 是美丽的，当且仅当 $1 \leq x, y, z \leq n, xy \leq n, yz \leq n, xz \leq n$ ，其中 n 是给定的正整数。

小 U 想为难一下小 C ，于是他向小 C 提出了 T 个问题，每个问题包含一个正整数 n ，你需要帮小 C 回答有多少个美丽的三元组。

输入格式

第一行包含一个整数 T 。

以下 T 行，每行一个整数 n ，表示一个问题。

输出格式

T 行，每行一个整数，表示对应问题的答案。

样例输入 1

```
3
1
2
5
```

样例输出 1

```
1
4
17
```

样例解释

对于第 1 组数据，满足条件的三元组为 $(1, 1, 1)$ 。

对于第 2 组数据，满足条件的三元组为 $(1, 1, 1), (1, 1, 2), (1, 2, 1), (2, 1, 1)$ 。

样例输入 2

见下发文件中的 `ex_triple2.in`。该样例满足数据点 3,4 的限制。

样例输出 2

见下发文件中的 `ex_triple2.out`。该样例满足数据点 3,4 的限制。

样例输入 3

见下发文件中的 `ex_triple3.in`。该样例满足数据点 5,6 的限制。

样例输出 3

见下发文件中的 `ex_triple3.out`。该样例满足数据点 5,6 的限制。

样例输入 4

见下发文件中的 `ex_triple4.in`。该样例满足数据点 9,10 的限制。

样例输出 4

见下发文件中的 `ex_triple4.out`。该样例满足数据点 9,10 的限制。

数据范围

令 $N = \max n$ 。

数据点编号	附加限制
1,2	$N \leq 100$
3,4	$N \leq 2 \times 10^3$
5,6	$N \leq 5 \times 10^5$
7,8	$N \leq 10^7$
9,10	无

对于所有数据，保证 $T \leq 100, 1 \leq n \leq 10^9$ 。

B. 不是第一题

小 R 喜欢玩 Florr.io。他有 n 个花瓣槽位，每个槽位上的花瓣种类为 a_i ，需要满足 $1 \leq a_i \leq 3 \times 10^5$ 且 $a_i \in \mathbb{Z}$ 。

为了最大化他的攻击力，小 R 想让他的任意两个相邻槽位上花瓣种类之积互不相等。

在此基础上，小 E 想让小 R 花瓣的种类数尽量少。

形式化的，你需要构造整数序列 a ，使得 $\forall 1 \leq i \leq n, 1 \leq a_i \leq 3 \times 10^5$ ，并使得 a 序列中的不同元素个数最少。
 $\forall 1 \leq i < j \leq n - 1, a_i a_{i+1} \neq a_j a_{j+1}$

输入格式

一行一个整数 n 。

输出格式

第一行一个数 m ，表示 a 序列中不同元素的个数。

第二行 n 个数，表示 a_i 。

注意下发的输出文件中不会包含真实的 a 序列输出，只会包含 a 序列中不同元素个数的最小值。

样例输入 1

```
5
```

样例输出 1

```
3
998 244 353 353 998
```

样例解释

可以验证样例输出合法。

显然还有其他满足条件的输出，选手只需要输出任意一种即可获得分数。

样例输入 2

见下发文件中的 `ex_array2.in`。该样例满足子任务 2 的限制。

样例输出 2

见下发文件中的 `ex_array2.ans`。该样例满足子任务 2 的限制。

样例输入 3

见下发文件中的 `ex_array3.in`。该样例满足子任务 3 的限制。

样例输出 3

见下发文件中的 `ex_array3.ans`。该样例满足子任务 3 的限制。

样例输入 4

见下发文件中的 ex_array4.in 。该样例满足子任务 4 的限制。

样例输出 4

见下发文件中的 ex_array4.ans 。该样例满足子任务 4 的限制。

样例输入 5

见下发文件中的 ex_array5.in 。该样例满足子任务 8 的限制。

样例输出 5

见下发文件中的 ex_array5.ans 。该样例满足子任务 8 的限制。

数据范围与提示

本题包含自定义校验器 (spj) 。

本题评分方式如下：

1.如果你输出的 a_i 不满足 $1 \leq a_i \leq 3 \times 10^5$ ，或你输出的 m 与 a 序列中的不同元素个数不同，得 0 分；

2.如果你输出的 a_i 不满足 $\forall 1 \leq i < j \leq n, a_i a_{i+1} \neq a_j a_{j+1}$ ，但你输出的 m 与真实的最小值相同，你可以获得该测试点 20% 的分数。若你输出的 m 与真实最小值不同，得 0 分。

3.如果你输出的 a_i 满足上述所有条件，设你的构造中 a_i 的不同元素个数为 A ， a_i 中不同元素个数的最小值为 B ，则你可以获得该测试点 $x\%$ 的分数。

若 $A > 2B$ ， $x = 20$ ；若 $B + 1 < A \leq 2B$ ， $x = 40$ ；若 $A = B + 1$ ， $x = 60$ ；若 $A = B$ ， $x = 100$ 。

我们在下发文件中下发了本题的 checker，用于检测你构造方案的合法性。

注意下发的输出文件中不会包含真实的 a 序列输出，只会包含 a 序列中不同元素个数的最小值。

Checker 的使用方式：在对应目录下打开终端，输入命令 checker [input] [output] [answer] 。

其中 input 表示输入文件名称，output 表示你的输出文件名称，answer 表示答案文件名称。

一个可能的命令为 "checker ex_array1.in array.out ex_array1.ans" 。

本题为子任务评测，每个子任务的得分为其中所有测试点得分的 **最小值**。

Subtask ID	附加限制	分数
1	$n \leq 10$	5
2	$n \leq 20$	10
3	$n \leq 10^3$	15
4	$n \leq 10^5$	10
5	$n \in [10^5, 10^5 + 100]$	15
6	$n \in [1.5 \times 10^5, 1.5 \times 10^5 + 100]$	15
7	$n \leq 3 \times 10^5$	5

Subtask ID	附加限制	分数
8	$n \leq 10^6$	25

C. 第三题

小 L 有一棵 n 个节点的树，其中有一些节点是关键节点。

小 M 会在树上选择一些**互不相交**的联通块，需要保证每个关键节点都至少被一个联通块包含。

若这些联通块的大小分别为 $c_1, c_2 \dots c_k$ ，定义这些联通块的权值为 $\sum_{i=1}^k (c_i + T)$ 。

你需要对 $l \leq T \leq r$ ，求出权值的最小值。可以参考样例解释进行理解。

输入格式

第一行包含三个数 n, l, r 。

第二行包含 n 个数 t_i ，若 $t_i = 1$ ，则点 i 为关键节点，否则 i 不为关键节点。

以下 $n - 1$ 行，每行 2 个数 u, v ，表示一条边。

输出格式

$r - l + 1$ 个数，第 i 个数表示当 $T = i + l - 1$ 时的答案。

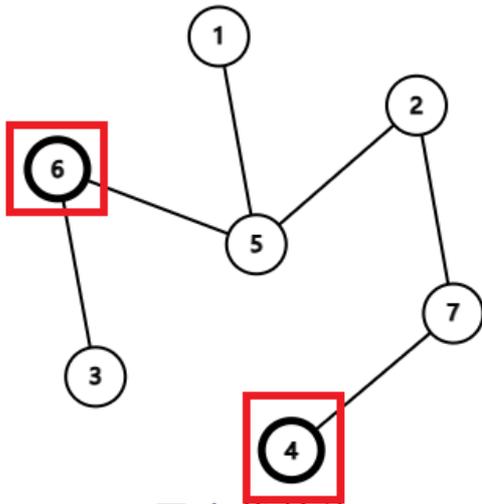
样例输入 1

```
7 1 4
0001010
7 4
5 6
7 2
5 1
6 3
2 5
```

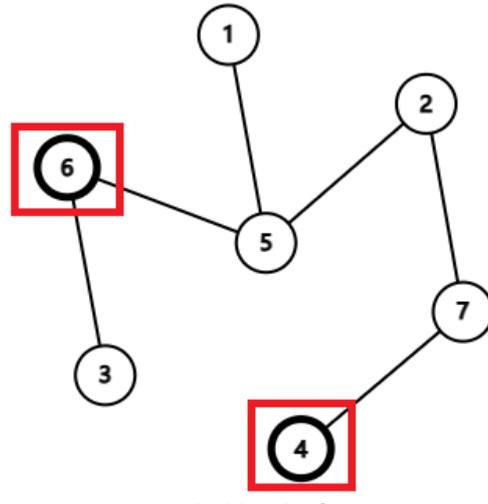
样例输出 1

```
4
6
8
9
```

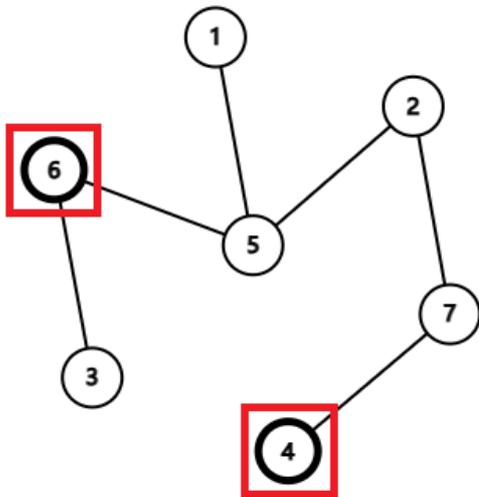
样例解释



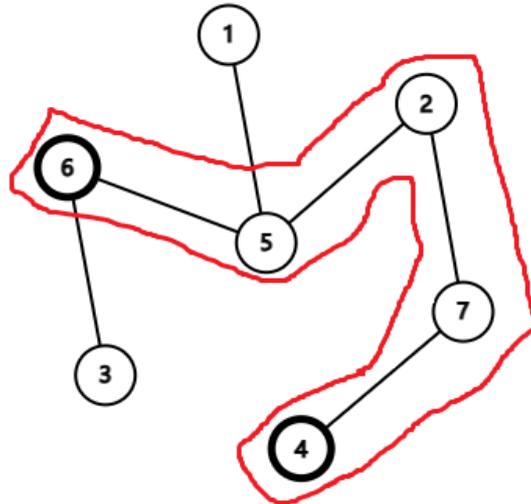
T=1 最小代价为 4



T=2 最小代价为 6



T=3 最小代价为 8



T=4 最小代价为 9

样例输入 2

见下发文件中的 ex_price2.in 。该样例满足数据点 1,2,3 的限制。

样例输出 2

见下发文件中的 ex_price2.out 。该样例满足满足数据点 1,2,3 的限制。

样例输入 3

见下发文件中的 ex_price3.in 。该样例满足数据点 4,5,6 的限制。

样例输出 3

见下发文件中的 ex_price3.out 。该样例满足数据点 4,5,6 的限制。

样例输入 4

见下发文件中的 ex_price4.in 。该样例满足数据点 9 的限制。

样例输出 4

见下发文件中的 ex_price4.out 。该样例满足数据点 9 的限制。

样例输入 5

见下发文件中的 ex_price5.in 。该样例满足数据点 10,11 的限制。

样例输出 5

见下发文件中的 ex_price5.out 。该样例满足数据点 10,11 的限制。

样例输入 6

见下发文件中的 ex_price6.in 。该样例满足数据点 16,17,18,19 的限制。

样例输出 6

见下发文件中的 ex_price6.out 。该样例满足数据点 16,17,18,19 的限制。

样例输入 7

见下发文件中的 ex_price7.in 。该样例满足数据点 20,21,22,23,24,25 的限制。

样例输出 7

见下发文件中的 ex_price7.out 。该样例满足数据点 20,21,22,23,24,25 的限制。

数据范围

数据点编号	限制
1,2,3	$n \leq 20$
4,5,6	$n \leq 500$
7,8,9	$n \leq 3 \times 10^3$
10,11	$n \leq 10^4$
12,13	$R \leq 10^3$
14,15	$L \geq 10^5$
16,17,18,19	$n \leq 10^5$
20,21,22,23,24,25	无

对于所有数据，保证 $n \leq 2 \times 10^5, 1 \leq l \leq r \leq n$ 。

另外，数据点 3,6,9,19,25 满足 $\forall 1 \leq i < n$, 点 i 与点 $i + 1$ 连边。

D. 不是第三题

小 T 生活在 ω 国。 ω 国有 n 个城市，由于 ω 国已经高度发达，每两个城市间都有一座双向的时空隧道。连接 i, j 的时空隧道的通行时间为 $t_{i,j}$ 。

生活在 Ω 国的小 S ，想在假期去看望小 T 。但由于一些经济原因，小 S 只能申请到 $n - 1$ 条时空隧道的使用权。小 S 想让这 $n - 1$ 条时空隧道连通整个 ω 国。

同时, 小 S 会在 u 点进入 ω 国。由于时空隧道的特殊性, 从 i 到 j 的通行时间 $w_{i,j}$ 为: 设 i 到 j 不经过同一座城市两次的路径为 i, p_1, \dots, p_k, j , 则 $w_{i,j} = \min(t_{i,p_1}, \min_{i=1}^{k-1} t_{p_i, p_{i+1}}, t_{p_k, j})$ 。简要说, 就是 i 到 j 在生成树上的唯一简单路径上边权的最小值。

注意此时只有有使用权的 $n - 1$ 条时空隧道可以使用, 因此, 路径是唯一的。

对于一种使用权的方案, 定义其代价为 $\sum_{i=1}^n w_{u,i}$ 。小 S 想让你对于所有起点 u , 求出此时所有方案中代价的最小值。

输入格式

第一行包含一个整数 n 。

以下 $n - 1$ 行中, 第 i 行包含 $n - i$ 个数。

第 i 行的第 j 个数表示 $t_{i, i+j}$ 。

输出格式

n 个数, 第 i 行的数表示起点 i 的答案。

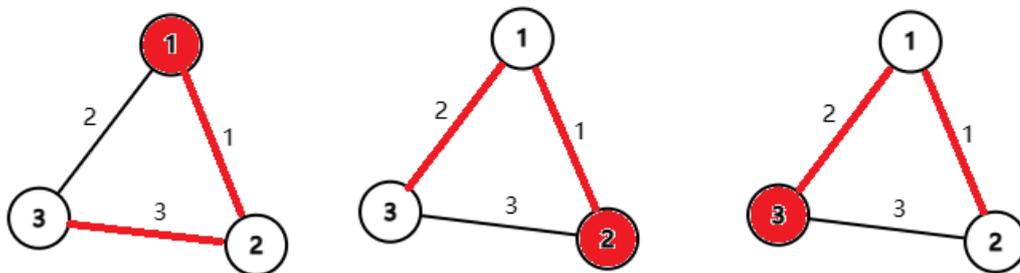
样例输入1

```
3
1 2
3
```

样例输出1

```
2
2
3
```

样例解释



红点表示起点, 红色的边表示有使用权的边。 $i = 1, 2, 3$ 的最小代价分别为 2, 2, 3。

样例输入 2

见下发文件中的 `ex_travel2.in`。该样例满足数据点 1 的限制。

样例输出 2

见下发文件中的 ex_travel2.out 。该样例满足数据点 1 的限制。

样例输入 3

见下发文件中的 ex_travel3.in 。该样例满足数据点 6 的限制。

样例输出 3

见下发文件中的 ex_travel3.out 。该样例满足数据点 6 的限制。

样例输入 4

见下发文件中的 ex_travel4.in 。该样例满足数据点 9 的限制。

样例输出 4

见下发文件中的 ex_travel4.out 。该样例满足数据点 9 的限制。

样例输入 5

见下发文件中的 ex_travel5.in 。该样例满足数据点 16,17,18,19 的限制。

样例输出 5

见下发文件中的 ex_travel5.out 。该样例满足数据点 16,17,18,19 的限制。

数据范围与提示

测试点编号	限制
1,2	$n \leq 6$
3,4	$n \leq 9$
5,6	$n \leq 15$
7,8,9	$n \leq 100$
10	$t_{i,j} = 1$
11,12	若 $j = i + 1$ 则 $t_{i,j} = 1$, 否则 $t_{i,j} = 10^9$
13,14,15	$n \leq 10^3$
16,17,18,19,20	$n \leq 2 \times 10^3$

对于所有数据, 保证 $n \leq 2 \times 10^3, 1 \leq t_{i,j} \leq 10^9$ 。

另外, 数据点 2,4,6,9,15,20 保证 $t_{i,j} \leq n$ 。

本题输入规模较大, 建议选手使用关流 cin 或快速读入。