2024 年第十届中国大学生程序设计竞赛（重庆）

时间：2024 年 11 月 10 日 09:00 ~ 14:00

目 录

A . 乘积，欧拉函数，求和

【题目描述】

给定 n 个数 $a_1, a_2, \cdots, a_n$，你需要求以下式子的值：

$$\sum_{S \subseteq \{1,2,\cdots,n\}} \varphi\left(\prod_{i \in S} a_i\right).$$

其中 φ 为欧拉函数，$\varphi(x)$ 表示在 $[1, x]$ 内与 x 互质的整数数量，例如

- $\varphi(6) = 2$，因为在 $[1, 6]$ 内有 1 和 5 与 6 互质。
- $\varphi(1) = 1$，因为在 $[1, 1]$ 内有 1 与 1 互质。

另外，我们定义 $\prod_{i \in \varnothing} a_i = 1$。

答案可能很大，你需要求出其对质数 998244353 取模的结果。

【输入格式】

从标准输入读入数据。

输入的第一行为一个整数 n $(1 \leq n \leq 2000)$ 表示数的数量，接下来一行 n 个整数 $a_1, a_2, \cdots, a_n$ $(1 \leq a_i \leq 3000)$。

【输出格式】

输出到标准输出。

输出一行一个整数表示答案，对 998244353 取模。

【样例 1 输入】

```
3
1 2 3
```

【样例 1 输出】

```
12
```

【样例 1 解释】

共有八种 S 的选择，所有选择得到的 $\prod_{i \in S} a_i$ 分别为 $1, 1, 2, 2, 3, 3, 6, 6$。可以计算得到 $\varphi(1) = \varphi(2) = 1, \varphi(3) = \varphi(6) = 2$，因此答案为 $1 \times 4 + 2 \times 4 = 12$。

B . osu!mania

【题目描述】

osu! 是一款风靡全球的音乐游戏，分为四个模式：osu!，osu!taiko，osu!catch，osu!mania。osu!mania 是一款下落式节奏游戏，像是钢琴模拟器一样。这个模式主要由 woc2006 开发并移植。它基于各种轨道式音乐游戏（例如劲舞革命 (Dance Dance Revolution) 和狂热节拍 (Beatmania)）。

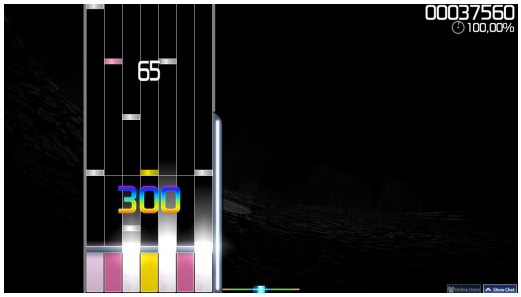


图 1: osu!mania 游戏界面

osu!mania 的每一局游戏都有若干个音符（note）组成。玩家游玩过程中，对每个音符的打击都会得到一个判定结果，分为 MAX、300、200、100、50、MISS（0）。记整局游戏中，玩家得到这些判定结果的音符个数分别为 a, b, c, d, e, f，则该局游戏的准确率（Accuracy，以下简称 Acc）可以按如下方式计算：

$$\text{Acc} = \frac{300a + 300b + 200c + 100d + 50e + 0f}{300(a + b + c + d + e + f)} \times 100\%.$$

由于准确率可能为无限小数，游戏中将会显示其四舍五入的结果。具体地，玩家每局游戏的准确率将会被四舍五入，并保留百分数形式下的两位小数，即精确到 10^{-4}。

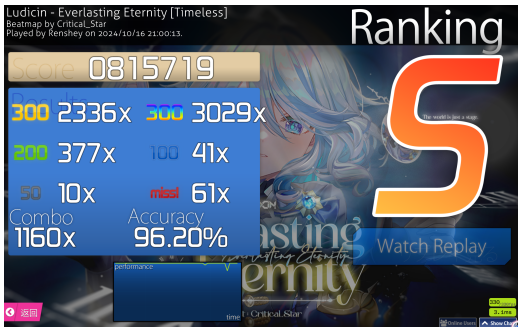


图 2: Acc 计算示例

例如，在上图所示的一局游戏中，玩家的准确率为

$$\begin{aligned}\text{Acc} &= \frac{300 \times 3029 + 300 \times 2336 + 200 \times 377 + 100 \times 41 + 50 \times 10 + 0 \times 61}{300 \times (3029 + 2336 + 377 + 41 + 10 + 61)} \\ &= \frac{16895}{17562} \approx 96.20\%.\end{aligned}$$

除准确率外，osu!mania 中还有另一个重要的衡量单曲游玩成绩的指标——个人表现（Personal Performance，以下简称 pp）。个人表现与谱面星数与判定成绩相关。谱面星数会决定在该谱面可以获得的 pp 上限，记为 ppmax。玩家在一局游戏中获得个人表现可以按如下方式计算（a, b, c, d, e, f 的含义与 Acc 计算方式中相同）：

$$\text{pp} = \max\left(0, \frac{320a + 300b + 200c + 100d + 50e + 0f}{320(a + b + c + d + e + f)} - 80\%\right) \times 5 \times \text{ppmax}.$$

玩家每局游戏的个人表现将会被四舍五入到最接近的整数。例如，在图 2 所示的成绩中，若 $\text{ppmax} = 663$，则玩家的个人表现为

$$\text{pp} = \max\left(0, \frac{1750080}{1873280} - 80\%\right) \times 5 \times 630 = \frac{1237635}{2927} \approx 423.$$

给定一个谱面的 pp 上限 ppmax 与玩家在该谱面上的一局游戏的判定结果 a, b, c, d, e, f。求玩家该局游戏的准确率与个人表现。

【输入格式】

从标准输入读入数据。

本题包含多组测试数据。

输入的第一行包含一个正整数 T，表示数据组数。保证 $1 \le T \le 100$。

对于每组测试数据：

输入的第一行包含一个非负整数 ppmax。保证 $0 \le \text{ppmax} \le 3000$。

输入的第二行包含六个非负整数 a, b, c, d, e, f，含义如题目描述所示。保证 $0 \le a, b, c, d, e, f \le 2 \times 10^4$ 且 $a + b + c + d + e + f \ge 1$。

【输出格式】

输出到标准输出。

对于每组测试数据：输出一行两个数，以空格隔开。其中第一个数表示准确率，以百分数形式输出，精确到 10^{-4}；第二个数为个人表现，以整数形式输出。

【样例 1 输入】

```
2
630
3029 2336 377 41 10 61
3000
20000 10000 0 0 0 0
```

【样例 1 输出】

```
96.20% 423
100.00% 2688
```

【样例 1 解释】

在样例的第二组测试数据中，准确率为 100%，个人表现为

$$\text{pp} = \max\left(0, \frac{320 \times 20000 + 300 \times 10000}{320 \times (20000 + 10000)} - 80\%\right) \times 5 \times 3000 = 2687.5 \approx 2688.$$

C . 连方

【题目描述】

给定正整数 n 和两个仅包含字符 . 和 # 的长度为 n 的字符串 a, b，请构造一个 $7 \times n$ 的仅包含字符 . 与 # 的矩阵，满足以下条件：

- 矩阵第 1 行与 a 相同，第 7 行与 b 相同。
- 由四方向连通的 # 构成的图形均为**实心的**矩形。具体地：
  - 对于两个 # 字符，如果可以从其中一个 # 字符出发，在有限步之内仅经过 # 字符到达另一个 # 字符，其中每一步均为向**上、左、下、右四个方向**之一移动一格，则称这两个 # 字符在同一组。那么，由同一组内的所有 # 字符构成的图形均为**实心的**矩形。
- 所有的 # 字符八方向连通，具体地：
  - 对于任意两个 # 字符，均可以从其中一个 # 字符出发，在有限步之内仅经过 # 字符到达另一个 # 字符，其中每一步均为向**上、左、下、右、左上、右上、左下、右下八个方向**之一移动一格。

请输出任意一个满足条件的矩阵，或判定无解。

【输入格式】

从标准输入读入数据。

输入的第一行包含一个正整数 T $(1 \le T \le 10^4)$，代表数据组数。

每组数据第一行包含一个正整数 n $(2 \le n \le 10^5)$，代表矩阵的宽度。

接下来两行分别包含仅包含字符 . 与 #，长度为 n 的两个字符串 a, b，代表矩阵的第 1 行与第 7 行。

保证 a 与 b 均包含至少一个 #。

保证单个测试点内所有 n 的总和不超过 2×10^5。

【输出格式】

输出到标准输出。

对于每组测试数据，如果不存在满足要求的矩阵，则输出一行一个字符串 `No`。

如果存在满足要求的矩阵，则先输出一行一个字符串 `Yes`，然后输出 7 行，每行包含一个长度为 n 的字符串，代表你构造的矩阵。

【样例 1 输入】

```
4
4
#..#
.##.
5
##.#.
.#.##
6
######
.####.
27
.######.######.####.#.#####
.####...####..#.......#####
```

【样例 1 输出】

```
Yes
#..#
.##.
.##.
#..#
.##.
.##.
.##.
Yes
##.#.
##.#.
##.#.
..#..
.#.##
.#.##
.#.##
No
Yes
.######.######.####.#.#####
#......#......#....#.#.....
```

```
#......#......#....#.#.....
#......#.......####..#.....
#......#......#......#.....
#......#......#......#.....
.####...####..#.......#####
```

【样例 1 解释】

对于第一组数据，以下是另一种正确答案：

```
#..#
#..#
#..#
.##.
#..#
#..#
.##.
```

但以下矩阵不是正确答案，因为由 $(1,1)(2,1)(2,2)$ 这组四方向连通的 # 字符构成的图形不是矩形。对于 $(4,1)(4,2)(5,1)(6,1)$ 这组四方向连通的 # 字符同理。

```
#..#
##.#
..#.
##.#
#.#.
#..#
.##.
```

同样，以下矩阵也不是正确答案，因为位于 $(1,1)$ 的 # 字符与其他 # 字符不满足八方向连通的条件。

```
#..#
...#
...#
...#
...#
...#
.##.
```

D . 有限小数

【题目描述】

给定两个互质正整数 a, b，你需要求两个非负整数 c, d，满足以下两个条件：

- $\frac{a}{b} + \frac{c}{d}$ 为十进制下的整数或有限小数。
- $1 \le d \le 10^9$。

在所有满足条件的非负整数对 (c, d) 中，请求出 c 最小的一对。

一个有理数 x 是十进制下的有限小数，当且仅当将 x 在十进制下以小数形式写出后，小数点后的位数是有限的，即存在正整数 k，整数 p 和整数数组 $(q_1, q_2, \ldots, q_k)$ 满足 $0 \le q_i \le 9$，使得 $x = p + \sum_{i=1}^{k} q_i \cdot 10^{-i}$。

【输入格式】

从标准输入读入数据。

第一行包含一个正整数 T $(1 \le T \le 10^4)$，表示数据组数。

每组数据包含一行两个正整数 a, b $(1 \le a \le b \le 10^6)$，含义如题目描述所示。保证 $\gcd(a, b) = 1$。

【输出格式】

输出到标准输出。

对于每组数据，输出一行两个非负整数 c, d。如果有多组正确答案，输出任意一组即可。

【样例 1 输入】

```
4
1 2
2 3
3 7
19 79
```

【样例 1 输出】

```
0 1
1 3
1 14
3 316
```

【样例 1 解释】

对于第一组数据，由于 $\frac{1}{2} = 0.5$ 是有限小数，因此输出 (c, d) 满足 $c = 0$ 且 $1 \le d \le 10^9$ 即可。

对于第二组数据，$\frac{2}{3} + \frac{1}{3} = 1$ 是整数，且 $\frac{2}{3} = 0.666\ldots$ 不是有限小数，因此 $c = 1$ 是最小可能值。

对于第三组数据，$\frac{3}{7} + \frac{1}{14} = \frac{1}{2} = 0.5$ 是有限小数。

对于第四组数据，$\frac{19}{79} + \frac{3}{316} = \frac{1}{4} = 0.25$ 是有限小数，且可以证明不存在 $0 \le c \le 2$，$1 \le d \le 10^9$ 使得 $\frac{19}{79} + \frac{c}{d}$ 是有限小数。

E . 合成大西瓜

【题目描述】

小白有 n 个西瓜（保证 n 是奇数），每个西瓜有个重量 a_i。她在这 n 个西瓜之间建立了 m 条无向边，使任意两个西瓜之间都至少存在一条路径能到达。

小白现在可以选择三个西瓜进行合并，具体地，她会选择三个不同的西瓜 x, y, z 满足 x, y 之间有一条无向边，y, z 之间有一条无向边。她会得到一个新的西瓜 w，其重量 $a_w = \max(a_y, \min(a_x, a_z))$。接下来，她对于“至少和 x, y, z 中某个西瓜之间有无向边”的西瓜 t，建立了一条 (w, t) 之间的无向边。最后，小白删去了 x, y, z 三个西瓜以及某一端为 x, y, z 的无向边。

可以证明一定存在一种合并 $\frac{n-1}{2}$ 次的方案使得最后仅剩下一个西瓜，小白想知道最后那个西瓜重量的最大值是多少。

【输入格式】

从标准输入读入数据。

第一行两个非负整数 n, m。保证 $1 \le n \le 10^5$，$0 \le m \le 10^5$，且 n 是奇数。

第二行 n 个正整数 $a_1, a_2, \ldots, a_n$，表示每个西瓜的重量。保证 $1 \le a_i \le n$。

接下来 m 行，每行两个正整数 x, y 表示图上的一条无向边 (x, y)。保证 $1 \le x, y \le n$ 且 $x \ne y$。

保证给定的无向图连通，且无重边与自环。

【输出格式】

输出到标准输出。

一行一个正整数，表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
2
```

【样例 2 输入】

```
1 0
1
```

【样例 2 输出】

```
1
```

F . Pico Park

【题目描述】

Menji 在玩一款多人益智小游戏。

在这个游戏中，有 n 名玩家，依次站在数轴的 $1, 2, 3, \ldots, n$ 处，第 i 名玩家有一个面向的方向 d_i，为向左或向右。

每名玩家手里有一把缩小枪，玩家会按照一个排列 p 的顺序行动，当轮到玩家 x 行动时：

- 若该玩家已经被缩小，则其不会进行任何行动。
- 否则，其会向其面对的方向发射子弹，子弹会击中面对方向的**第一个未被缩小**的玩家（若面对方向已经没有玩家，则不会击中任何人）。被击中的玩家会立刻被缩小。

由于形势混乱，在实际游戏中，p 会在所有 $n!$ 个可能的排列中随机选取。

Menji 想知道，对于每一个 $1 \le k \le n$，有多少个排列会使**最终剩余 k 个未被缩小的玩家**？

由于答案很大，你只需要输出答案对 998244353 取模后的值。

【输入格式】

从标准输入读入数据。

第一行一个整数 n $(2 \le n \le 500)$。

接下来一行一个长度为 n 的字符串 s。其中 $s_i \in \{\mathrm{L}, \mathrm{R}\}$，若 $s_i = \mathrm{L}$ 则第 i 名玩家面向左方（即玩家 1 所在方向），若 $s_i = \mathrm{R}$ 则第 i 名玩家面向右方（即玩家 n 所在方向）。

【输出格式】

输出到标准输出。

输出一行 n 个数，其中第 i 个数表示最终剩余 i 个玩家的排列数。

【样例 1 输入】

```
2
RL
```

【样例 1 输出】

```
2 0
```

【样例 2 输入】

```
4
LLRR
```

【样例 2 输出】

```
0 24 0 0
```

【样例 3 输入】

```
10
LRLRLLRRRR
```

【样例 3 输出】

```
0 0 0 604800 3024000 0 0 0 0 0
```

G．魔弹

【题目描述】

Menji 在玩一款休闲养成游戏。

在这个游戏中，有 n 名员工，依次站在数轴的 $1, 2, 3, \ldots, n$ 处，第 i 名员工有一个面向的方向 d_i，为向左或向右。

每名员工手里有一把名为魔弹的武器，玩家会按照一个排列 p 的顺序行动，当轮到玩家 x 行动时：

- 若该员工已经倒下，则其不会进行任何行动。
- 否则，其会向其面对的方向发射子弹，子弹会击中面对方向的**所有未倒下**的玩家（若面对方向已经没有玩家，则不会击中任何人）。被击中的员工会立刻倒下。

由于形势混乱，在实际游戏中，p 会在所有 $n!$ 个可能的排列中随机选取。

Menji 想知道，对于每一个 $1 \le k \le n$，有多少个排列会使**员工 k 最终没有倒下**？

由于答案很大，你只需要输出答案对 998244353 取模后的值。

【输入格式】

从标准输入读入数据。

第一行一个整数 n $(2 \le n \le 10^5)$。

接下来一行一个长度为 n 的字符串 s。其中 $s_i \in \{\mathrm{L}, \mathrm{R}\}$，若 $s_i = \mathrm{L}$ 则第 i 名员工面向左方（即员工 1 所在方向），若 $s_i = \mathrm{R}$ 则第 i 名员工面向右方（即员工 n 所在方向）。

【输出格式】

输出到标准输出。

输出一行 n 个数，其中第 i 个数表示最终剩余 i 个员工的排列数。

【样例 1 输入】

```
2
RL
```

【样例 1 输出】

```
1 1
```

【样例 2 输入】

```
4
LLRR
```

【样例 2 输出】

```
0 24 24 0
```

【样例 3 输入】

```
4
RLRL
```

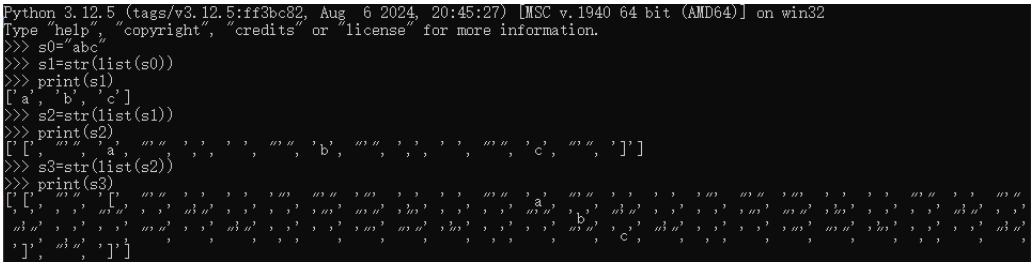
【样例 3 输出】

```
9 6 6 9
```

H . str(list(s))

【题目描述】

在 Python 语言中，若将一个字符串先转换为序列 list，再转换为字符串 str 类型，可以得到一个新的字符串。以下控制台运行结果描述了这一过程。



以下我们形式化地描述从 s 到 str(list(s)) 的变换。为了得到 str(list(s))，我们需要将 s 中的每个字符 s_i $(0 \le i < |s|)$ 替换为使用如下规则产生的长度为 5 的字符串 $t_{i,0}t_{i,1}t_{i,2}t_{i,3}t_{i,4}$：

- $t_{i,2} = s_i$。
- 当 s_i 不是单引号 ' 时，$t_{i,1}$ 和 $t_{i,3}$ 均为单引号 ' (ASCII 39)，否则 $t_{i,1}$ 和 $t_{i,3}$ 均为双引号 " (ASCII 34)。
- 当 $i \neq 0$ 时，$t_{i,0}$ 为空格 ' ' (ASCII 32)，否则为左中括号 [(ASCII 91)。
- 当 $i \neq |s| - 1$ 时，$t_{i,4}$ 为逗号 ,(ASCII 44) ，否则为右中括号] (ASCII 93)。

现在，输入一个由除空白字符以外的可见字符（即 ASCII 码 33 至 126 的所有字符）构成的字符串 s，设 $s^0 = s$，对于整数 $i > 0$，定义 $s^i = \text{str}(\text{list}(s^{i-1}))$。再输入两个整数 k 和 p，你需要对每个 $0 \le j < p$ 求出 s^k 中所有下标模 p 为 j 的字符的 ASCII 码的和，字符串下标从 0 开始编号。若不存在下标模 p 为 j 的字符，认为答案为 0。答案可能很大，你需要将答案对 $(10^9 + 7)$ 取模。

【输入格式】

从标准输入读入数据。

输入的第一行一个由除空白字符以外的可见字符构成的字符串 s $(1 \le |s| \le 10^5)$，第二行两个整数 k, p $(1 \le k, p \le 3,000)$。

【输出格式】

输出到标准输出。

输出一行 p 个整数，第 $(j+1)$ 个整数表示 s^k 中下标模 p 为 j 的所有字符的 ASCII 码的和，对 $(10^9 + 7)$ 取模。

【样例 1 输入】

```
abc
1 16
```

【样例 1 输出】

```
91 39 97 39 44 32 39 98 39 44 32 39 99 39 93 0
```

【样例 1 解释】

该组样例的最终字符串 s^1 即为题目描述的 Python 控制台运行过程中的 s1。

【样例 2 输入】

```
abc
2 7
```

【样例 2 输出】

```
472 420 580 408 474 439 429
```

【样例 2 解释】

该组样例的最终字符串 s^2 即为题目描述的 Python 控制台运行过程中的 s2。

【样例 3】

见题目目录下的 ***3.in*** 与 ***3.ans***。

【样例 3 解释】

该样例的字符串中包含了所有除空白字符以外的可见字符。

I . 算术

【题目描述】

Menji 学习了加法和乘法。

Menji 有一些写着 $1 \sim 9$ 的卡片，其中写着 i 的有 a_i 张。

Menji 每次会选择两张卡片，并选择将他们的和或者他们的积写在一张新的卡片上，之后他会丢弃选择的两张卡片，并拿起新的一张卡片。

可以发现，经过 $\left(\sum_{i=1}^{9} a_i\right) - 1$ 轮操作之后，Menji 手上只剩下一张卡片，Menji 想要最大化这张卡片上数字的值，但由于卡片数量太少，Menji 无法独立完成这个任务，希望你能帮他求出最后的数字最大能是多少。

由于本题答案很大，你只需要输出答案对 998244353 取模后的值。**注意，你需要输出的是最大值** $\bmod 998244353$**，而不是** $\bmod 998244353$ **意义下的最大值。**

【输入格式】

从标准输入读入数据。

本题含有多组测试数据。

第一行一个正整数 $T(1 \le T \le 1000)$，表示数据组数。

之后 T 行，每行 9 个非负整数 $a_1, a_2, \ldots a_9$ $(0 \le a_i \le 100, \sum_{i=1}^{9} a_i \ge 1)$。

【输出格式】

输出到标准输出。

输出 T 行，其中第 i 行是第 i 组数据中最终剩余的数的最大值对 998244353 取模的结果。

【样例 1 输入】

```
7
5 3 0 0 0 0 0 0 0
4 1 1 1 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 2
99 88 77 66 55 44 33 22 11
100 90 80 70 60 50 40 30 20
```

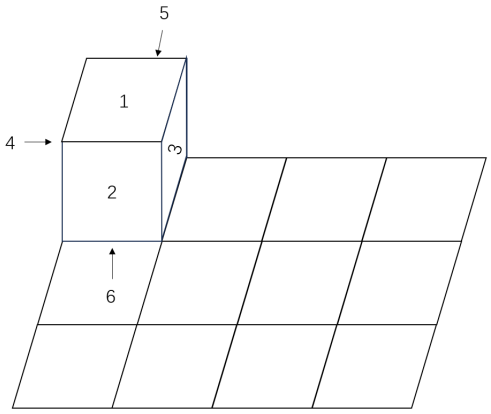
【样例 1 输出】

```
54
108
1
10
90
90553232
143532368
```

J . 骰子

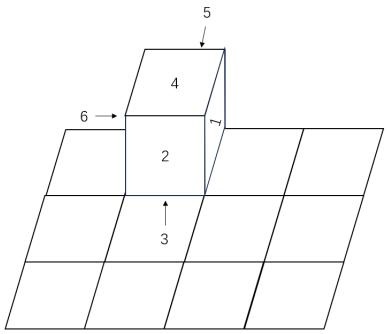
【题目描述】

在 n 行 m 列的网格的最左上角的格子上，有一个边长和网格格子边长相等的骰子。初始，这个骰子 1 在顶面，2 朝前，3 朝右，i 的背后是 $7-i$，如下图所示。



现在你可以做任意多次操作，每次操作为以下两种：

- 若当前骰子所在的格子没有数字，在这个格子上写下骰子底面的数字；
- 选择上下左右四个方向的某一个，将骰子沿着这个方向滚一次：选择骰子底面对应方向的棱，将骰子沿着这条棱旋转九十度。下图展示了初始状态向右滚一次的结果。你不能将骰子滚出网格。



注意：你可以在骰子经过一个没有数字的格子时选择不在这个格子上写下骰子底面的数字。

你希望最大化最后网格上所有写过数字的格子的数字的和。

【输入格式】

从标准输入读入数据。

输入一行两个整数 n, m $(2 \leq n, m \leq 10^3)$，表示网格的长和宽。

【输出格式】

输出到标准输出。

输出一行一个整数，表示在进行任意多次操作后，网格上所有写过数字的格子的数字的和的最大值。

【样例 1 输入】

```
2 2
```

【样例 1 输出】

```
24
```

K . 小 C 的神秘图形

【题目描述】

对于正整数 n，用如下方法定义一个 $3^n \times 3^n$ 的 01 矩阵 A_n：

- 若 $3^{n-1} \le i < 2 \times 3^{n-1}$ 或者 $3^{n-1} \le j < 2 \times 3^{n-1}$，则

$$A_n(i,j) = \begin{cases} 1, & n = 1, \\ A_{n-1}(i \bmod 3^{n-1}, j \bmod 3^{n-1}), & n \ge 2. \end{cases}$$

  其中 $x \bmod y$ 表示 x 对 y 取模后的结果；

- 否则，$A_n(i,j) = 0$。

其中，$A_n(i,j)$ 表示矩阵 A_n 第 i 行第 j 列的元素，并且行、列的编号均从 0 开始。

现在，给定正整数 n，小 C 有两个长度为 n 的数字串，其中每位都是 $0,1,2$ 中的一个，代表了两个三进制数 n_1, n_2（可能包含前导 0）。你需要帮小 C 求出 $A_n(n_1, n_2)$ 的值。

【输入格式】

从标准输入读入数据。

第一行输入一个正整数 $n(1 \le n \le 10^5)$，含义见题目描述。

接下来两行，每行输入一个长度为 n 的数字串，分别表示三进制数 n_1, n_2。

【输出格式】

输出到标准输出。

输出一个整数，表示 $A_n(n_1, n_2)$ 的值。

【样例 1 输入】

```
2
20
01
```

【样例 1 输出】

```
0
```

【样例 1 解释】

事实上，$n = 2$ 时有

$$A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

而输入的数字串分别对应 $n_1 = 6, n_2 = 1$，进而 $A_2(n_1, n_2) = 0$。

【样例 2 输入】

```
3
102
011
```

【样例 2 输出】

```
1
```

L . 沙堆

【题目描述】

给定一棵 n 个点的无向树。初始每个点 i $(1 \le i \le n)$ 有点权 c_i。考察如下操作：

- 设 $\deg_x$ 为点 x 的度数。若存在一个点 x 满足 $c_x \ge \deg_x$，则选择任意一个满足该条件的点 x，将 c_x 减去 $\deg_x$ 并将 x 的所有邻居的权值加 1。

称一个点权序列 $(c'_1, \cdots, c'_n)$ 为**终态**当且仅当以上操作无法进行，即所有点 y 均满足 $c'_y < \deg_y$。

可以证明，对于任意无向树和点权序列，只有以下两种可能的情况：

1. 无论如何操作，在有限次操作之后都会得到终态，且任意操作均会得到同一个终态。
2. 无论如何操作，都无法在有限次操作后得到终态。

你需要判断给定的初始状态属于以上哪一种情况。如果属于第一种情况，则你需要给出任意进行操作能够得到的唯一的终态。

【输入格式】

从标准输入读入数据。

输入的第一行包含一个正整数 n $(1 \le n \le 10^6)$ 表示树的点数。

接下来 $n-1$ 行每行两个整数 x, y $(1 \le x, y \le n)$ 表示树的一条边。

接下来一行 n 个整数 c_i $(0 \le c_i \le 10^9)$ 表示初始点权。

【输出格式】

输出到标准输出。

如果在有限次操作内无法到达终态，输出 `-1`，否则输出一行 n 个整数，依次描述终态每个点的点权。

【样例 1 输入】

```
6
1 2
2 3
2 4
1 5
4 6
1 1 0 0 1 1
```

【样例 1 输出】

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

【样例 1 解释】

考察以下操作序列：

- 对 6 进行操作，得到点权序列 $(1,1,0,1,1,0)$；
- 对 5 进行操作，得到点权序列 $(2,1,0,1,0,0)$；
- 对 1 进行操作，得到点权序列 $(0,2,0,1,1,0)$；
- 对 5 进行操作，得到点权序列 $(1,2,0,1,0,0)$。

而点权序列 $(1,2,0,1,0,0)$ 是终态，故输出 `1 2 0 1 0 0`。

【样例 2 输入】

```
12
1 2
1 3
2 4
3 5
5 6
2 7
7 8
4 9
8 10
5 11
3 12
2 0 0 0 1 0 1 0 1 1 0 1
```

【样例 2 输出】

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

【样例 3】

见题目目录下的 ***3.in*** 与 ***3.ans***。

M . Median Replacement

【题目描述】

给定一个长为 n 的整数序列 $a_1, a_2, \ldots, a_n$，你需要对 a 进行若干次如下操作，使得 a 中所有数均相等：

- 选择一段长为**大于** 1 **的奇数**的区间 $a_l, a_{l+1}, \ldots, a_r$，并将此区间内的**所有数**均替换为它们的中位数。

设最终 $a_1 = a_2 = \ldots = a_n = x$，我们定义序列 a 的值为 x 的最大值。

请你求出所有满足 $\forall\, 1 \le i \le n$，$l_i \le a_i \le r_i$ 的整数序列 a 的值之和。

由于答案可能很大，请对 $10^9 + 7$ 取模。

【输入格式】

从标准输入读入数据。

第一行一个整数 T，表示测试数据组数。

对于每组测试数据：

- 第一行一个整数 n。
- 第二行 n 个整数 $l_1, l_2, \ldots, l_n$。
- 第三行 n 个整数 $r_1, r_2, \ldots, r_n$。

保证 $1 \le T \le 10$，$3 \le n \le 150$，$1 \le l_i \le r_i \le 10^9$。

【输出格式】

输出到标准输出。

对于每组测试数据，输出一行一个整数表示答案对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

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

【样例 1 输出】

```
1
5
```

【样例 1 解释】

对于第一组测试数据，a 只能为 $[1,1,1]$，值为 1，故答案为 1。

对于第二组测试数据，a 可以为 $[1,1,1],[1,1,2],[1,2,1],[1,2,2]$，值分别为 $1,1,1,2$，故答案为 5。