

A 拼图

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

Joky_02 收到了他诛神之战的礼物-----一个纯金色的拼图。

这个拼图由 n 种不同边数的多边形构成，其中第 i 种多边形是 e_i 边形。

为了方便拼图，Joky_02 把拼图块先按边数进行了分类。接着，他发现， e_i 边形的拼图碎片每条边上要么没有东西，要么有一个凸起，要么有一个凹陷。并且，拼图制作精良，根本无法区分出正反面。

同时，Joky_02 注意到，这个拼图没有任何两条相邻的边同时含有突起，或者同时含有凹陷。

没有办法，为了更好地进行拼图，Joky_02 必须对每个多边形的拼图碎片进行进一步的细分。他需要对每个 e_i 边形的拼图碎片，进一步按形状划分。直到剩下的部分是因为无法区分正反面，也无法区分旋转，从而看起来一样的。

他想知道，这样划分过后，他的拼图碎片最多会有多少类呢？

输入格式：

第一行一个整数 $n(1 \leq n \leq 10^5 - 2)$ ，表示拼图碎片的的多边形种数。

第二行一个整数 $e_i(3 \leq e_i \leq 10^5)$ ，表示第 i 种多边形的边数，保证任意两个多边形的边数互不相同。

输出格式

一行一个整数，表示拼图碎片最多的类数对 998244353 取模的结果。

输入样例：

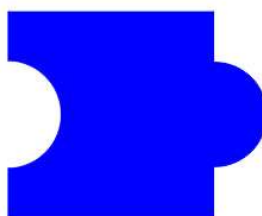
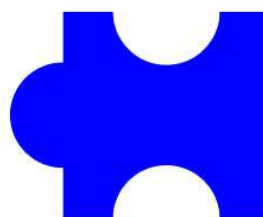
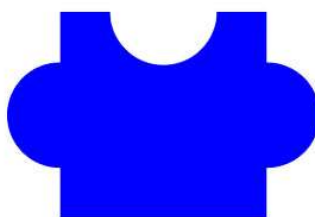
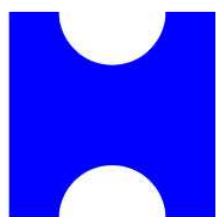
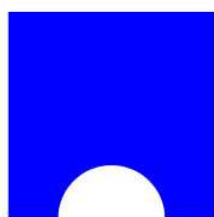
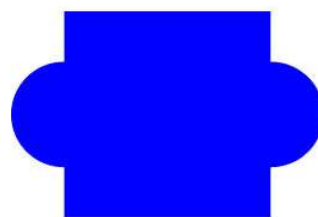
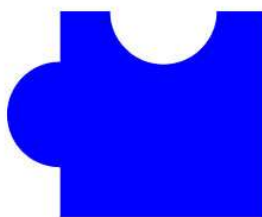
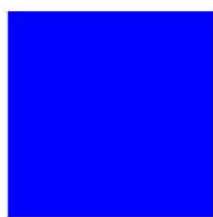
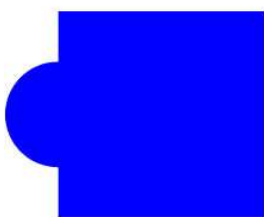
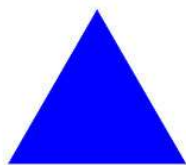
```
2
3 4
```

输出样例：

```
14
```

Notes:

以下为满足题意的三角形拼图与四边形拼图。且可以证明，其余所有其他三角形与四边形拼图都是不满足题意，或与以下拼图之一看起来一样。



B FFT

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

定义欧拉数 $\left\langle n \atop k \right\rangle$ 是 $\{1, 2, 3, \dots, n\}$ 的有 k 个升高的排列 $\pi_1 \pi_2 \pi_3 \cdots \pi_n$ 的个数，也就是说，在其中 k 个地方有 $\pi_j < \pi_{j+1}$ 。

欧拉数具有 $\left\langle n \atop m \right\rangle = \sum_{k=0}^m \binom{n+1}{k} (-1)^k (m+1-k)^n$ 的性质。

其求法可以为 $\left\langle n \atop m \right\rangle = (n-m) \left\langle n-1 \atop m-1 \right\rangle + (m+1) \left\langle n-1 \atop m \right\rangle$ 。

即考虑强制从小到大放数：如果放在最前面，升高的排列数不会增加；如果放在之前某个升高的排列中间，那么会减少一个排列增加一个排列，总数不变；否则放在其他的位置升高的排列数都会增加一个。

现在给定一个 n ，stump 想知道 $\sum_{i=0}^n \left\langle n \atop i \right\rangle$ 。

输入格式：

一行一个数 $n (1 \leq n \leq 10^7)$ 。

输出格式

一行一个整数表示 $\sum_{i=0}^n \left\langle n \atop i \right\rangle$ 对 998244353 取模的结果。

输入样例：

3

输出样例：

6

C 魔法师

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

YahAha最近刚收到魔法学院的邀请，正式入学成为一名见习魔法师，勤奋好学的他从导师那里学来了 n 个咒语。

但他作为一个低年级的见习魔法师，魔力的水平太低，常规的使用咒语发挥不出效果，于是他找导师学来了一个融合魔咒。

现在YahAha可以同时使用两个咒语，并释放更大的魔力，但这两个咒语被使用后会消耗掉。

发动魔咒的咒语是一个由字母组成的字符串。假设有两个咒语 S_1 和 S_2 ，设 $f(S_1, S_2)$ 表示两个字符串的最长公共前缀的长度， $g(S_1, S_2)$ 表示两个字符串的最长公共后缀的长度， $Len = \min(f(S_1, S_2), g(S_1, S_2))$ ，两个咒语 S_1, S_2 同时使用释放的魔力为 Len^2 。

现在YahAha想用这种方法使用掉他的 n 个咒语，他想知道他能释放的最大魔力值。

输入格式：

第一行一个整数 $n(1 \leq n \leq 2 \times 10^5)$ 表示YahAha会的咒语数量。

接下来 n 行，每行一个字符串，第 $i + 1$ 行表示YahAha掌握的第 i 个咒语。咒语仅由大小写字母组成，所有咒语长度和不超过 3×10^5 。

输出格式

一行一个整数，表示YahAha能释放的最大魔力值

输入样例：

```
3
GDCPC
GDP
GCC
```

输出样例：

```
1
```

D 剪纸

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

Amber 收到了母星屏障顺利启动的纪念品-----一张纸。

她可以自己决定这张纸的初始大小：长为 r ，宽为 l ，并且 $l, r \leq n$ 。

她觉得一张纸太少了，所以她获得这张纸后打算按照这样的规则开始剪纸：每次剪去一个最大的正方形，直到纸张被剪完。

她希望在剪完之后，不同的正方形数量越多越好。

由于她还要忙着去帮助其他文明，所以想让你帮她确认初始的纸张大小。

为了减少浪费，她需要你输出和最小的一对 l, r ，保证答案唯一。

输入格式：

第一行一个整数 $n(1 \leq n \leq 10^{18})$ 。

输出格式

一行两个整数 $l, r(l \leq r)$ 。

输入样例：

3

输出样例：

2 3

E 黑白大陆

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

毕业出了魔法学院的YahAHa回到了他的家乡-----黑白大陆。

黑白大陆是一块由 $n \times m$ 个格子组成的方形大陆，在每一个格子上标有黑色或白色，在黑白大陆上，不同颜色的相邻格子之间时常会发生战争。

热爱和平的YahAHa想要把黑白大陆统一成象征和平的白色。

YahAHa会一种神奇的魔法，他可以把连通的白色区域变成黑色，连通的黑色区域变成白色。

YahAHa不想消耗太多法力，于是他想知道，把整个黑白大陆变成白色最少需要使用几次魔法。

输入格式：

第一行两个整数 n, m ($1 \leq n, m \leq 50$)表示黑白大陆的大小。

接下来 n 行，每行 m 个整数 $a_{i,j}$ 表示黑白大陆每一个格子的颜色。 ($0 \leq a_{i,j} \leq 1$,其中0表示代表和平的白色，1表示黑色)

输出格式

一行一个整数，表示把整个黑白大陆变成白色最少需要使用几次魔法。

输入样例：

```
3 3
1 0 1
0 1 0
1 0 1
```

输出样例：

```
3
```

F 望舒客栈的每日委托

代码长度限制 16 KB

Java (javac)

时间限制 6000 ms

内存限制 512 MB

其他编译器

时间限制 3000 ms

内存限制 512 MB

今天望舒客栈又迎来了新的客人。

客人们可以分为 n 批，每一批有 x_i ($1 \leq x_i \leq 4$) 个人，并且有着不同的抵达时间 a_i 和离开时间 d_i 。

望舒客栈有 m 张四人桌从左到右排开，现在每批到达望舒客栈的客人会从左到右选择最近的合适的四人桌落座，但是由于每批人性格不同，选座的方式也不一样。社交恐惧症会选择第一张空的四人桌落座，而社交牛逼症会选择第一张能坐下 x_i 个人的四人桌落座，当然他们不会管这张四人桌上有没有可怜的社交恐惧症，且所有人落座后便不再换位置。

现在望舒客栈的老板非尔戈黛特想知道 m 至少为多少时，才能保证所有的客人都能落座。

输入格式：

第一行输入一个整数 n ($1 \leq n \leq 10^6$)。

接下来 n 行，每行输入 4 个整数 x_i, a_i, d_i, t_i ，分别表示第 i 批客人的人数，抵达时间，离开时间，为社交恐惧症还是社交牛逼症。 $t_i = 0$ 时，表示这批人是社交恐惧症； $t_i = 1$ 时，表示这批人是社交牛逼症。保证每批人的离开时间一定严格迟于到达时间， $1 \leq a_i, d_i \leq 2n$ ，且所有的 $2n$ 个时刻保证互不相同。

输出格式

一个整数 m 表示答案。

输入样例：

```
4
1 1 8 0
3 2 3 0
1 4 6 0
2 5 7 0
```

输出样例：

```
3
```

G Rock&Frog

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

青蛙喜欢摇滚，他认为跳跃和橘子汽水就是摇滚。

现在有 n 个位置编号 $1 \sim n$ ，在 n 号位置上有橘子汽水。

青蛙一开始在 1 号位置上，它想要通过跳跃一次或跳跃多次跳到 n 号位置上。

青蛙可以且仅可以从一个位置 i 跳到另一个位置 j 上 ($i < j$)，花费 $a_i \times c_j^2 + b_i \times c_j$ 的代价。

青蛙觉得这样不够摇滚，为此它添加了一个限制：**保证**任意 $1 \leq i, j \leq n$ ， $a_i > a_j$ 仅当 $b_i \geq b_j$ ， $b_i > b_j$ 仅当 $a_i \geq a_j$ 。

值得注意的是，代价可能为负的，在这种情况下青蛙认为自己被倒贴请喝橘子汽水。

它想知道喝到橘子汽水的最小代价。

输入格式：

第一行输入一个整数 n ($2 \leq n \leq 10^5$)。

接下来 n 行，第 i 行读入三个整数 a_i, b_i, c_i ($-10^9 \leq a_i, b_i \leq 10^9, 0 \leq c_i \leq 10^9$)。

保证任意 $1 \leq i, j \leq n$ ， $a_i > a_j$ 仅当 $b_i \geq b_j$ ， $b_i > b_j$ 仅当 $a_i \geq a_j$ 。

输出格式

输出一个整数 ans ，表示最小代价。

输入样例：

```
6
1 2 4
3 4 3
-1 0 12
3 10 4
-3 -2 1
6 11 3
```

输出样例：

```
-30
```


H 梅花易数

代码长度限制	16 KB
Java (javac)	
时间限制	2000 ms
内存限制	512 MB
其他编译器	
时间限制	1000 ms
内存限制	512 MB

Alice 很喜欢梅花易数。他学习了梅花易数中的年月日起卦法。

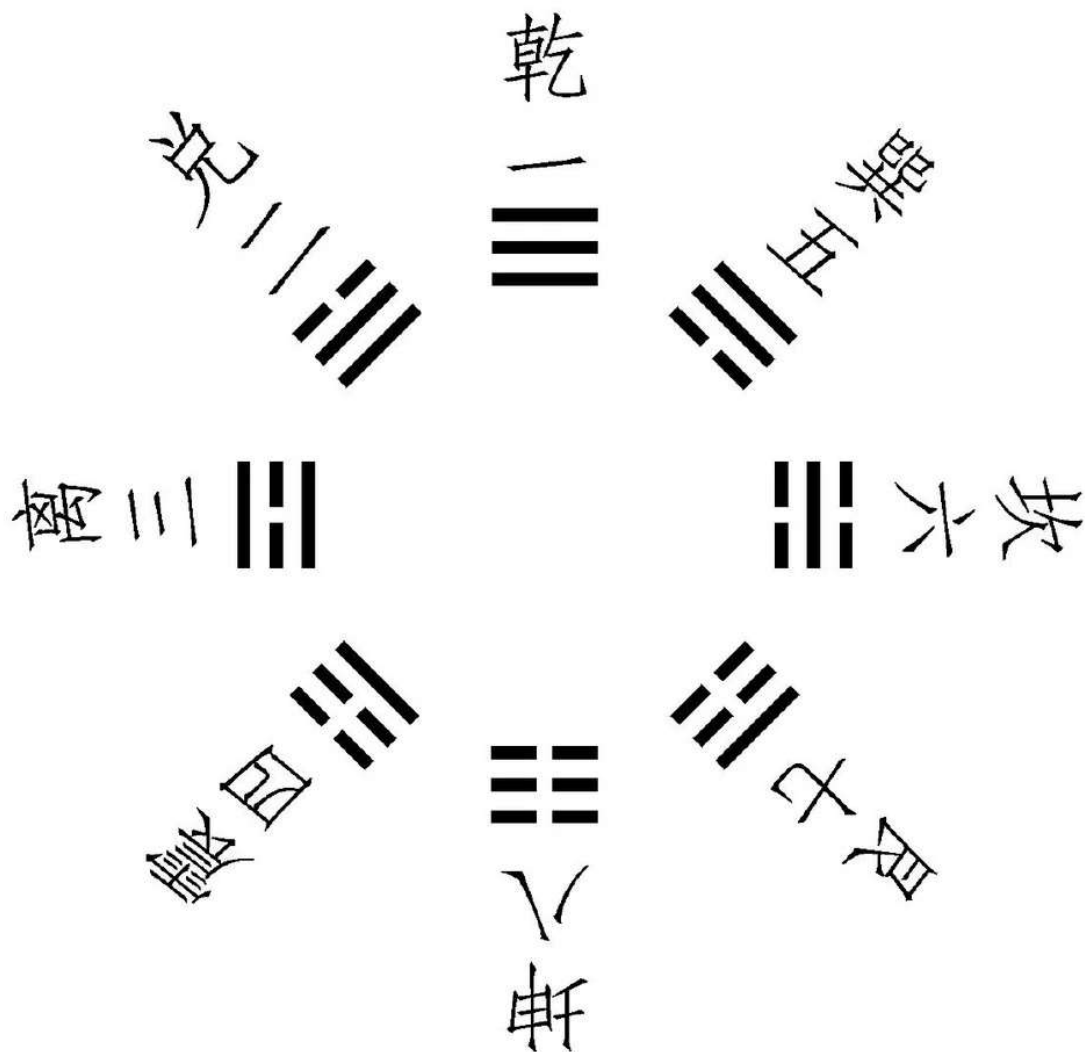
按照年月日起卦法，起卦前需要知道某个时间对应的年月日，以及该时间对应的时辰。

对于年份 y ，我们知道子年(Zi)数一、丑年(Chou)数二、寅年(Yin)数三、卯年(Mao)数四、辰年(Chen)数五、巳年(Si)数六、午年(Wu)数七、未年(Wei)数八、申年(Shen)数九、酉年(You)数十、戌年(Xu)数十一、亥年(Hai)数十二。

对于月 m 与日 d ，即以农历对应的月、日为准。

而对于时辰 h ，当时间位于子、丑、寅、卯、辰、巳、午、未、申、酉、戌、亥时辰时，其对应数字同年份一致。

根据梅花易数。起卦时，将 $y + m + d$ 除以 8 的余数作为上卦（当余数为 0 时，认为余数为 8）。上卦由三条爻竖直组成，每条爻要么是阳爻（由一条连续横线构成），要么是阴爻（由两条横线和中间的空格构成）。当余数分别为 1 到 8 时，对应的卦象形态分别为乾（☰）、兑（☱）、离（☲）、震（☳）、巽（☴）、坎（☵）、艮（☶）、坤（☷）。



随后，将 $y + m + d + h$ 除以 8 的余数作为下卦（同样，当余数为 0 时，认为余数为 8），其对应卦象与上卦相同。

将这两步的上卦与下卦拼接起来，即能得到本卦。

而梅花易数又将 $y + m + d + h$ 除以 6 的余数作为动爻（当余数为 0 时，认为余数为 6）。于是，将本卦从下往上数的对应爻翻转（即阴爻改为阳爻，阳爻改为阴爻）即可得到变卦。

现在，作为 Alice 的"好基友"，你需要帮他实现一个输入年、月、日、时，直接得出本卦和变卦图像的程
序。为了方便实现，我们不妨假设 Alice 不会输入不合法的日期。

输入格式：

输入一行包括四个部分组成，分别表示 y, m, d, h 。

其中 y, h 由十二地支的形式给定，保证其必为题中所给的十二种字符串之一。

而 m, d 以整数的形式给定，其中保证 $1 \leq m \leq 12$ 且 $1 \leq d \leq 31$ 。

输出格式

输出由 13 行构成，前 6 行为本卦，第 7 行为空行，后 6 行为变卦。

当卦象从上往下数的当前位为阴爻时，请在一行内输出"短线-空格-短线"的形式；当为阳爻时，请输出"短线-短线-短线"的形式。具体形式见样例。

请注意，不要输出行末空格。

输入样例：

Chen 12 17 Shen

输出样例：

--

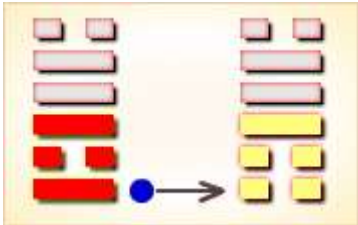
--

--

--
--

Notes:

取辰为5，申为9。
故上卦 $5+12+17=34$ ，34除以8余2。得到上卦为兑"☱"。
而下卦 $5+12+17+9=43$ ，43除以8余3。得到下卦为离"☲"。
将上下卦拼接即得到本卦。
而动爻数为 $43 \div 6 = 7 \dots 1$ ，故动爻为自下向上第一爻。将动爻翻转即得到变卦。



I Rock&String

代码长度限制 16 KB

Java (javac)

时间限制 6000 ms

内存限制 512 MB

其他编译器

时间限制 3000 ms

内存限制 512 MB

字符串就是摇滚。

小 J 喜欢字符串，他认为一个字符串 s 相对于 t 是摇滚的，当且仅当 s 是 t 的一个 *border*。

对于 *border* 的定义是：若 $0 \leq |s| < |t|$, s 是 t 的一个前缀且 s 是 t 的一个后缀, 则称 s 是 t 的一个 *border*。

现在，小 J 打算给你一个长度为 n 的字符串 $S[1..n]$ 和 m 次询问。

每次询问描述一个 S 的子串 $S[l..r]$, 求 $\min\{r' - l' + 1\}$, 满足 $1 \leq l' \leq r' \leq n$ 且 $S[l..r]$ 相对于 $S[l'..r']$ 是摇滚的。

输入格式：

第一行输入一个长度为 n ($1 \leq n \leq 2 \times 10^5$) 的字符串 S 。

第二行读入一个整数 m ($1 \leq m \leq 2 \times 10^5$), 表示询问次数。

接下来的 m 行, 每行包括两个整数 l, r ($1 \leq l \leq r \leq n$), 描述一个询问的子串 $S[l..r]$ 。

保证 S 中只包含小写字母。

输出格式

输出 m 行, 每行包含一个整数 ans , 表示对应询问的答案。若不存在 $S[l'..r']$ 使得 $S[l..r]$ 相对于 $S[l'..r']$ 是摇滚的, 则输出 '-1' (不包含引号)。

输入样例：

```
abaabccababc
5
1 2
3 4
6 6
1 9
1 3
```

输出样例：

```
4
-1
2
-1
10
```

J 新英雄

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

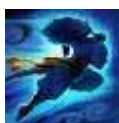
内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

新英雄索亚拥有法力值和两种移动方式。



1. 踏：索亚花费 3 秒移动到距离恰好为 1 的下一**友方**小兵上（敌方小兵**不能**作为移动目标），并获得 1 点法力值。



2. 斩：索亚花费 1 秒移动到距离恰好为 1 的下一小兵上（敌我小兵**都可以**作为移动目标），并消耗 1 点法力值，**法力值为 0 时不能用"斩"**。

现在设计师有一个问题：假设索亚现在站在一个友方小兵上，他前面有 n 个小兵排成笔直的一队，从近到远从 1 到 n 编号，起点位置的友方小兵视为第 0 个小兵。相邻两个小兵距离为 1。小兵分为我方小兵和敌方小兵。

一开始索亚的法力值为 0。索亚可以用"踏"向前或向后移动到一个友方的相邻小兵上；或者在有法力时，用"斩"向前或向后移动到一个任意的相邻小兵上。那么索亚到最后一个小兵最短需要多少时间？

输入格式：

输入第一行两个正整数 n, m ($1 \leq n \leq 10^9, 1 \leq m \leq 10^5$)。

接下来共 m 行，每行共三个正整数 t, l, r ($1 \leq t \leq 2, 1 \leq l, r \leq n$)。

对于每一行，当 $t = 1$ 时，代表第 l 个到第 r 个是友方小兵；当 $t = 2$ 时，代表第 l 个到第 r 个是敌方小兵。**保证每个整点有且只有一个小兵。**

输出格式

输出共一行，如果索亚无法移动到最后一个小兵，则输出 "0/21/0"（不含双引号）。
否则包含一个正整数，表示需要的最短时间。

输入样例 1：

```
8 2
1 1 5
2 6 8
```

输出样例 1:

16

输入样例 2:

8 2
1 1 1
2 2 8

输出样例 2:

28

输入样例 3:

1 1
2 1 1

输出样例 3:

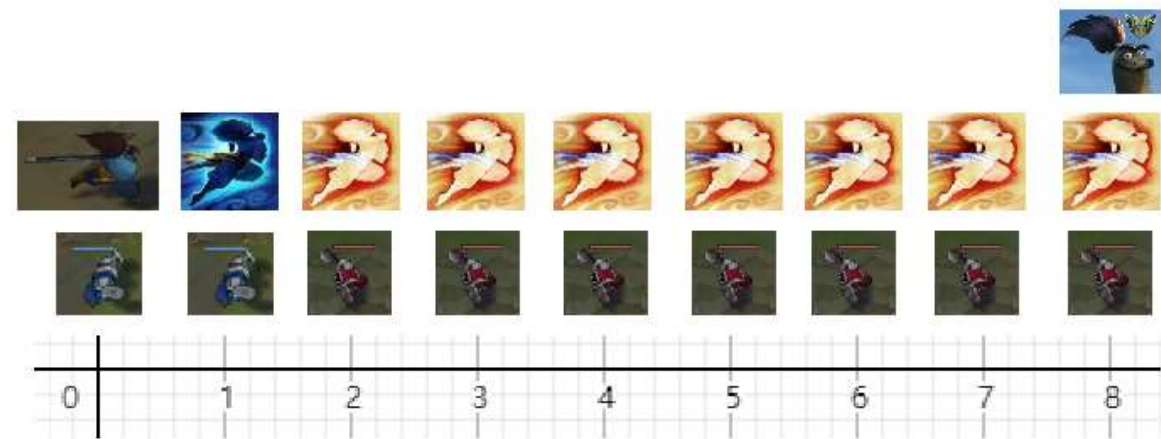
0/21/0

Notes:

如图所示，



样例1中，索亚按如图方式，最短时间为 $3 \times 4 + 1 \times 4 = 16$ 。



样例 2 中，索亚在 0 号点与 1 号点来回"踏"来积攒法力，最短时间为 $3 \times 7 + 1 \times 7 = 28$ 。



样例3中，因为索亚没有法力值，所以不能对第一个敌方小兵使用"斩"。

K 斐波那契

代码长度限制 16 KB

Java (javac)

时间限制 2000 ms

内存限制 512 MB

其他编译器

时间限制 1000 ms

内存限制 512 MB

给定一个 n 个点 m 条带权边的有向图和一个自然数 K ，有向图中包含重边和自环。

点集 $V = \{1, 2, \dots, n\}$ ，边集 $E = \{(u_1, v_1), (u_2, v_2), \dots, (u_m, v_m)\}$ 。

我们定义 $w_{(i,j)}$ ，表示 i 向 j 连一条有向边，并且权值为自然数 $w_{(i,j)}$ 。

现在，我们定义一条路径为 $(v_0, v_1, \dots, v_{K-1}, v_K)$ ， $\forall i$ ，有 $v_i \in V$ ， $(v_i, v_{i+1}) \in E$ ， v_i 表示依次经过的点，路径上依次经过的边有 $(v_0, v_1), (v_1, v_2), \dots, (v_{K-1}, v_K)$ ，注意这里点和边是可以重复的。

这条路径的收益为 $f(\sum_{k=1}^K w_{(v_{k-1}, v_k)})$ ，其中 $f(i)$ 表示斐波那契数列的第 i 项，

$f(0) = 0, f(1) = 1, f(i) = f(i-1) + f(i-2)$ 。

现在你需要对于所有的二元组 (i, j) 统计所有包含 K 条边的路径中，满足起点 $v_0 = i$ ，终点 $v_K = j$ 的路径的收益之和。

答案可能很大，你需要取模 998244353。

输入格式：

第一行输入三个数 $n(1 \leq n \leq 50), m(0 \leq m \leq 10^5), k(0 \leq k \leq 10^9)$ 。

第二行到第 $m+1$ 行，每行输入三个数 $u(1 \leq u \leq n), v(1 \leq v \leq n), w(0 \leq w \leq 10^9)$ ，表示 u 向 v 连一条有向边，边权为 w 。

输出格式

输出共 n 行，每行 n 个数。

第 i 行的第 j 个数表示起点 $v_0 = i$ ，终点 $v_K = j$ 的路径的收益之和。

注意：同一行的相邻两个数之间输出一个空格。

输入样例：

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

输出样例：

```
1288 411 809
665 212 432
0 0 5
```

L 启航者

代码长度限制 16 KB

Java (javac)

时间限制 4000 ms

内存限制 512 MB

其他编译器

时间限制 2000 ms

内存限制 512 MB

不要沉醉于这个虚幻的宁静长夜，夜幕终究会被打破

不要沉溺于你们温暖的摇篮，摇篮总有一天会倾覆

不要沉睡在心灵的庇护所中，心灵的庇护迟早会成为无法打破的枷锁

启程吧，在群星闪烁之前，启程吧，在长夜结束之前

再不快些出发，白昼就降临了

Amber 是启航者舰队中的一员，她被舰队派出执行一项任务，帮助一片星区中的文明。这片星区有 n 个星系，由 $n - 1$ 条超光速航道连通，每个星系中有 a_i 个文明。Amber 的飞船在进入一个星系后，就会派出无人机来帮助这个星系内所有的文明。

由于超光速航道的特性，每条航道只能被使用一次。

Amber 可以从任意一个星系开始执行任务。因为她没有学过算法，所以每次只会向航道还可以使用，相邻且文明数量最多的星系前进。

Amber 还要去研究魔力的统一波动猜想，所以找到了你，想让你帮助她计算一下，她应该从哪个星系开始，可以帮助到最多的文明，最多有多少个？

输入格式：

第一行一个整数 $n(1 \leq n \leq 10^6)$ 。

第二行 $n - 1$ 个整数 $f_1, f_2, \dots, f_{n-1}(1 \leq f_i \leq n - 1)$ 。其中 f_i 表示第 $i + 1$ 个星系和第 f_i 个星系之间存在一条超光速航道。

第三行 n 个整数 $a_1, a_2, \dots, a_n(0 \leq a_i \leq 10^9, \forall i \neq j, a_i \neq a_j)$ 。其中 a_i 表示每个星系的文明数量。

输出格式

第一行一个整数 x ，表示从第 x 个星系进入星区，如果有多个起点，选择编号最小的。

第二行一个整数 ans ，表示最多可以帮助 ans 个文明。

输入样例：

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

输出样例：

```
5
16
```

M 拉格朗日插值

代码长度限制 16 KB

Java (javac)

时间限制 6000 ms

内存限制 512 MB

其他编译器

时间限制 3000 ms

内存限制 512 MB

stump 有一个长度为 k 的数组 a , 他想知道满足 $\frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} + \cdots + \frac{x_k^2}{a_k^2} = 1$ 的条件下, $\prod_{i=1}^k x_i$ 的最大值。

现给定一个长度为 m 的数组 b , 随机从这 m 个数中选择 k 个数组成 a 数组, 求 $\prod_{i=1}^k x_i$ 的最大值的期望。

输入格式:

第一行给出两个整数 m, k ($1 \leq k \leq m \leq 10^5 \wedge k$ 是偶数), 分别代表 b 数组的长度和选择的数字个数。

接下来一行 m 个整数, 表示 b_1, b_2, \cdots, b_m ($0 < b_i < 10^9$)。

输出格式

一行一个整数表示 $\prod_{i=1}^k x_i$ 的最大值的期望对 998244353 取模的结果。

输入样例:

```
2 2
1 1
```

输出样例:

```
499122177
```