Problem A. Executable Log

时间限制: 2 seconds
空间限制: 512 megabytes

某天，小 W 在检查别人的服务器，他使用 `ls` 命令查看文件的时候，发现有一个 `.log` 文件（日志文件）竟然是绿色的（意味着该文件有执行权限）！

给出 n 个文件名和对应的 `rwx` 形式的读写权限，请计算这 n 个文件中有多少个 `.log` 文件有执行权限。

我们认为形如 "S`.log`" 的文件名为日志文件，其中 S 代表一个字符串（可以为空）。

"`rwx`" 分别代表读入权限、写入权限和执行权限，如果没有其中某个权限则会代替为 "`-`"，例如只有读入权限没有写入和执行权限为 "`r--`"。

Input

输入第一行为一个整数 n（$1 \le n \le 1000$），代表文件总数。

从第二行开始的 n 行，每行有两个字符串，分别代表文件名 f 和权限 p。

文件名 f 只包含英文字母、数字和字符 "`.`"，长度不超过 100。

权限 $p \in \{$"`---`","`r--`","`-w-`","`--x`","`rw-`","`r-x`","`-wx`","`rwx`"$\}$。

Output

输出一行一个整数，代表有执行权限的日志文件的个数。如果没有这样的文件请输出 0。

Example

standard input	standard output
5 .log rwx a.log.zip r-x proj.logger.cpp r-x main rwx b.log rw-	1

Problem B. 小 F 的分数

时间限制: 2 seconds
空间限制: 512 megabytes

MaiMai DX 是一款街机音乐游戏，在这个游戏中有 5 种基本判定：C.Perfect, Perfect, Great, Good, Miss；有 5 种音符：`TAP`, `HOLD`, `SLIDE`, `TOUCH`, `BREAK`。

在游玩时，小 F 恨透了 `HOLD`, `SLIDE`, `TOUCH`，于是创造了一个只有 `TAP` 和 `BREAK` 的版本。

游戏的分数是一个有 4 位小数的百分数，满分为 101.0000% ，分数是基础分数与奖励分数的和，向下保留到小数点后四位。

1. 基础分数（满分 100.0000%）

   基础分数由 `TAP` 和 `BREAK` 共同决定，`TAP` 和 `BREAK` 在不同判定评价下的得分如下表：

	C.Perfect	Perfect_1	Perfect_2	Great_1	Great_2	Great_3	Good	Miss
`TAP`	x	x	x	$0.8x$	$0.8x$	$0.8x$	$0.5x$	0
`BREAK`	$5x$	$5x$	$5x$	$4x$	$3x$	$2.5x$	$2x$	0

   其中 $x = \frac{100.0000\%}{T+5B}$，这里 T 和 B 分别表示 `TAP` 和 `BREAK` 的数量。

2. 奖励分数（满分 1.0000%）

   奖励分数只由 `BREAK` 决定，`BREAK` 在不同判定评价下的得分如下表：

	C.Perfect	Perfect_1	Perfect_2	Great_1	Great_2	Great_3	Good	Miss
`BREAK`	y	$0.75y$	$0.5y$	$0.4y$	$0.4y$	$0.4y$	$0.3y$	0

   其中 $y = \frac{1.0000\%}{B}$，这里 B 表示 `BREAK` 的数量。

请你来帮助小 F 计算一下游戏的分数吧。

Input

第一行包含 5 个整数 a_1, a_2, a_3, a_4, a_5（$0 \le a_i \le 10^3$），分别表示 `TAP` 音符 C.Perfect, Perfect, Great, Good, Miss 判定评价的数量。

第二行包含 8 个整数 $b_1, b_2, b_3, \cdots, b_8$（$0 \le b_i \le 10^2$），分别表示 `BREAK` 音符 C.Perfect, Perfect_1, Perfect_2, Great_1, Great_2, Great_3, Good, Miss 判定评价的数量。

数据保证 a_i 不全为 0 且 b_i 不全为 0。

Output

输出一个浮点数，表示游戏的分数，保留小数点后 4 位有效数字，不需要输出百分号 %。

Examples

standard input	standard output
2 1 1 0 0 1 0 1 0 0 0 0 0	99.3214
532 391 44 6 5 19 5 1 0 0 0 0 0	99.4068
1 0 0 0 1000 0 0 0 0 0 0 0 1	0.0994
1 0 0 0 0 1 0 0 0 0 0 0 0	101.0000

Note

在样例一中，`TAP` 音符获得了 2 个 C.Perfect 判定评价，1 个 Perfect 判定评价和 1 个 Great 判定评价，`BREAK` 音符获得了 1 个 C.Perfect 判定评价和 1 个 Perfect_2 判定评价。

`TAP` 的数量 $T = 2 + 1 + 1 + 0 + 0 = 4$，`BREAK` 的数量 $B = 1 + 0 + 1 + 0 + 0 + 0 + 0 + 0 = 2$。

因此根据公式，基础分数为 $\frac{13.8}{14} \times 100.0000\% \approx 98.5714285714\%$；

奖励分数为 $\frac{1.5}{2} \times 1.0000\% = 0.75\%$。

所以最终的游戏分数为 $98.5714285714\% + 0.7500\% = 99.3214285714\%$，向下保留到小数点后四位，结果为 99.3214%。

Problem C. 马拉车

时间限制: 2 seconds
空间限制: 512 megabytes

有 n 匹马和 m 辆车。

n 匹马分别编号为 $1 \sim n$，m 辆车分别编号为 $1 \sim m$。

第 i 匹马的耐力值为 E_i，第 i 辆车的重量为 W_i。

如果让第 i 匹马拉第 j 辆车，则这匹马的移动距离为 $\max(E_i - W_j, 0)$。

你需要为每一匹马分配**恰好**一辆车，且每辆车最多只能分配给一匹马。

你的目标是让所有马的移动距离中的**最小值**尽可能大。求出该最大值。

Input

第一行包含两个数 n 和 m（$1 \le n \le m \le 10^4$），分别表示马的数量和车的数量。

第二行包含 n 个数 $E_1, E_2, E_3, \cdots, E_n$（$1 \le E_i \le 10^9$），表示马的耐力值。

第三行包含 m 个数 $W_1, W_2, W_3, \cdots, W_m$（$1 \le W_i \le 10^9$），表示车的重量。

Output

输出一个整数，代表最小距离的最大值。

Example

standard input	standard output
3 4 2 3 2 1 3 2 1	1

Problem D. 简单树上问题

时间限制: 2 seconds
空间限制: 512 megabytes

给定一棵 n 个节点的有根树，节点编号为 1 到 n，根节点为 1。每个节点有一个非负整数权值，节点 i 的权值初始为 a_i。

你可以进行若干次操作。在一次操作中，你可以选择任意一个节点 u 和一个非负整数 x，随后将 u 及其子树中所有节点的权值**按位异或**[1]上[2] x，这次操作的代价为 siz_u 和 x 的乘积，其中 siz_u 表示以 u 为根的子树的大小（即节点个数，包括 u 本身）。

你的目标是通过若干次操作，将所有节点的权值变为 0。请你计算一下，代价的总和最小是多少?

Input

第一行一个整数 n（$2 \le n \le 10^5$），表示树的节点数。

第二行 n 个整数 $a_1, a_2, \cdots, a_n$（$0 \le a_i < 2^{30}$），表示每个节点的初始权值。

第三行 $n-1$ 个整数 $fa_2, fa_3, fa_4, \cdots, fa_n$（$1 \le fa_i < i$），分别表示节点 $2, 3, 4, \cdots, n$ 的父亲节点的编号。

输入数据保证这是一棵树。

Output

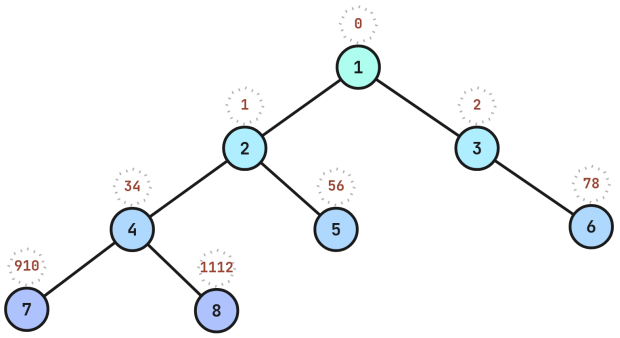
输出一个整数，表示将所有节点权值变为 0 的最小总代价。

Examples

standard input	standard output
8 0 1 2 34 56 78 910 1112 1 1 2 2 3 4 4	2333
3 1000000000 500000000 250000000 1 2	4608201600

Note

样例一中，树的形态如下图所示，点内显示的是点的编号，点上方的数字是该点的初始权值：



[1]按位异或是一种二进制按位运算，记作 $a \oplus b$。它对两个无符号整数的每一位进行运算：如果该位两个数相同，则该位结果为 0；如果不同，则结果为 1。例如 $5 \oplus 3 = (101)_2 \oplus (011)_2 = (110)_2 = 6$。

[2]「将 a 按位异或上 b」的意思是，把 a 变为 $a \oplus b$

Problem E. Colonization Assessment for Terraforming

时间限制: 2 seconds
空间限制: 512 megabytes

在公元 2230 年，地球联合国准备启动 CAT 计划（Colonization Assessment for Terraforming），在 A 星系寻找可能的殖民地。

该星系中有 n 个候选星球，第 i 颗星球的宜居度为 w_i。与此同时，还有 m 条双向航道连接这些星球，第 i 条航道连接 u_i 和 v_i 两颗星球。

你作为地球联合国科研人员，需要选择出 k 个星球，使得这组星球相互连通（即任意两个选择出的星球可以通过航道直接或间接到达，并且途中**不能**经过未选择的星球），使得这 k 个星球中**宜居度最小的星球的宜居度尽可能大**。

Input

第一行包含一个整数 T（$1 \le T \le 1000$）代表数据组数。对于每组数据:

- 第一行包含三个正整数 n, m, k（$1 \le k \le n \le 10^5, 0 \le m \le 10^5$）分别代表星球数、航道数和需要选择的星球数量。
- 下一行包含 n 个正整数 $w_1, w_2, w_3, \cdots, w_n$（$1 \le v_i \le 10^9$），分别表示这 n 颗星球的宜居度。
- 接下来的 m 行，第 i 行包含两个整数 u_i, v_i（$1 \le u_i, v_i \le n$），表示第 i 条航道连接的两颗星球。

保证多组数据的 n 之和与 m 之和均不超过 2×10^5。

Output

对于每组数据:

- 输出一行，包含一个整数，代表最小宜居度的最大值；如果无法选出 k 个可以相互到达的星球，请输出 -1。

Example

standard input	standard output
3 3 3 2 2 3 4 1 2 2 3 3 1 5 8 3 4 8 6 3 5 2 4 4 3 4 5 3 5 2 3 3 2 2 1 3 4 3 1 3 1 2 3 1 2	3 5 -1

Problem F. 迷宫 (I)

时间限制: 3 seconds
空间限制: 512 megabytes

小 i 最近了解到了一种用于生成“迷宫”的算法。它的具体流程如下：

首先准备一个大小为 $2^n \times 2^n$ 的正方形网格（n 为正整数）。第 i 行的第 j 个格点的坐标记为 (i, j)（$0 \le i, j < 2^n$）。

然后每个格点按照其 **Hilbert 序**依次编号为 0 到 $4^n - 1$。

坐标 (x, y) 的 **Hilbert 序** $H(n, x, y)$ 递归定义如下：

$$H(n,x,y) = \begin{cases} 0 & \text{if } n = 0 \\ H(n-1,\ y,\ x\ ) & \text{if } n \ge 1 \text{ and } 0 \le x, y < 2^{n-1} \\ H(n-1,\ x,\ y - 2^{n-1}\ ) + 4^{n-1} & \text{if } n \ge 1 \text{ and } 0 \le x < 2^{n-1} \text{ and } 2^{n-1} \le y < 2^n \\ H(n-1,\ x - 2^{n-1},\ y - 2^{n-1}\ ) + 2 \cdot 4^{n-1} & \text{if } n \ge 1 \text{ and } 2^{n-1} \le x, y < 2^n \\ H(n-1,\ 2^{n-1} - y - 1,\ 2^n - x - 1\ ) + 3 \cdot 4^{n-1} & \text{if } n \ge 1 \text{ and } 2^{n-1} \le x < 2^n \text{ and } 0 \le y < 2^{n-1} \end{cases}$$

以 $n = 3$ 的正方形网格为例，其中格点的 Hilbert 序如下图所示。

21 (0, 7)	22 (1, 7)	25 (2, 7)	26 (3, 7)	37 (4, 7)	38 (5, 7)	41 (6, 7)	42 (7, 7)
20 (0, 6)	23 (1, 6)	24 (2, 6)	27 (3, 6)	36 (4, 6)	39 (5, 6)	40 (6, 6)	43 (7, 6)
19 (0, 5)	18 (1, 5)	29 (2, 5)	28 (3, 5)	35 (4, 5)	34 (5, 5)	45 (6, 5)	44 (7, 5)
16 (0, 4)	17 (1, 4)	30 (2, 4)	31 (3, 4)	32 (4, 4)	33 (5, 4)	46 (6, 4)	47 (7, 4)
15 (0, 3)	12 (1, 3)	11 (2, 3)	10 (3, 3)	53 (4, 3)	52 (5, 3)	51 (6, 3)	48 (7, 3)
14 (0, 2)	13 (1, 2)	8 (2, 2)	9 (3, 2)	54 (4, 2)	55 (5, 2)	50 (6, 2)	49 (7, 2)
1 (0, 1)	2 (1, 1)	7 (2, 1)	6 (3, 1)	57 (4, 1)	56 (5, 1)	61 (6, 1)	62 (7, 1)
0 (0, 0)	3 (1, 0)	4 (2, 0)	5 (3, 0)	58 (4, 0)	59 (5, 0)	60 (6, 0)	63 (7, 0)

（不难发现这其实就是格点被二维希尔伯特曲线贯穿的顺序）

我们定义坐标 $(0, 0)$ 的格点为**起点**，坐标 $(2^n - 1, 0)$ 的格点为**终点**。

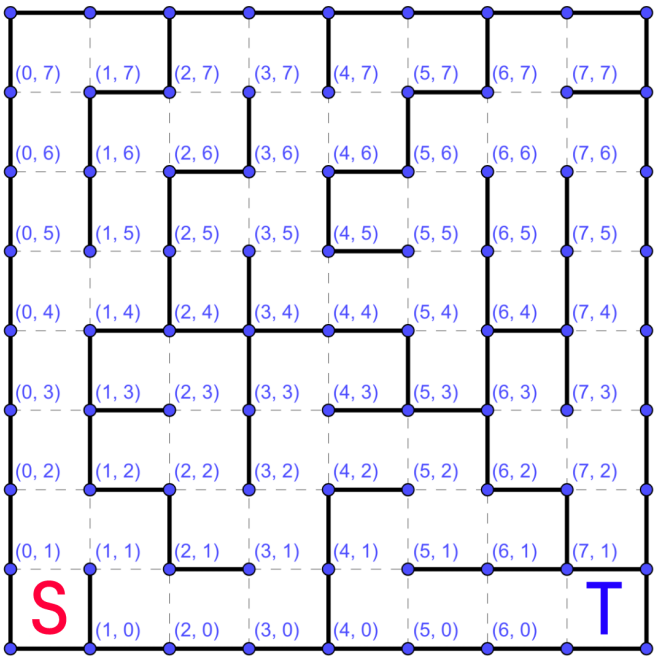
之后，我们在网格上进行连边操作。

具体的，对于**除起点外**的每个格点 (x, y)，我们将依次执行以下操作：

- 找到所有与 (x, y) **相邻**且 Hilbert 序**小于** $H(n, x, y)$ 的格点，构成集合 S。
- 从 S 中**等概率随机**选择一个格点 (x', y')，并在 (x, y) 和 (x', y') 之间连一条无向边。

这里所说的相邻定义为：如果两个格点 (x_1, y_1) 与 (x_2, y_2) 满足 $|x_1 - x_2| + |y_1 - y_2| = 1$，则我们称它们是**相邻**的。其中 $|x|$ 表示 x 的绝对值。

以下是一种可能的连边结果（两个相邻的格点之间没有实线分隔，当且仅当它们之间有连边）。



我们定义起点到终点的**距离**，为两者之间的最短路径所经过的格点数（包括起点和终点）。

小 i 想要知道，在连边操作结束后，起点到终点的距离的**期望**是多少？请你计算这个期望对 $998\,244\,353$ 取模的结果。

具体的，可以证明，答案可以被表示成一个最简分数 $\frac{p}{q}$（p 和 q 互质且 q 不被 $998\,244\,353$ 整除）。

你需要输出一个整数 x，满足 $0 \le x < 998\,244\,353$ 且 $x \cdot q \equiv p \pmod{998\,244\,353}$。

可以证明，在连边操作结束后，起点和终点一定连通。

Input

仅一行，包含一个正整数 n（$1 \le n \le 10$）。

Output

输出一行，包含一个整数，表示答案对 $998\,244\,353$ 取模的结果。

Examples

standard input	standard output
1	3
3	791800663

Problem G. 迷宫 (II)

时间限制: 4 seconds
空间限制: 512 megabytes

小 i 想知道，在一个长和宽都为 2^n 的正方形网格上，利用迷宫 (I) 中所述的连边操作，一共可能产生多少种不同的连边结果?

我们定义，两种连边结果是不同的，当且仅当存在一条边，使得它在其中一个连边结果中出现，而在另一个连边结果中没有出现。

由于这个数字可能很大，因此请你输出它对一个给定**正整数** m 取模后的结果。

Input

第一行一个整数 T（$1 \leq T \leq 10^4$），表示一共有 T 组数据。

对于每组数据:

- 仅一行，包含两个整数 n 和 m（$1 \leq n, m \leq 10^9$）。

Output

对于每组数据，输出一行，包含一个整数，表示答案。

Example

standard input	standard output
6 1 1 1 10 2 10000 3 999999999 114514 1919810 999999999 998244353	0 2 288 613014318 975328 98879783

Problem H. 子序列、达标率与期望

时间限制: 4 seconds
空间限制: 512 megabytes

在本次期末考试中，有大约一半的同学没有达到平均分，希望他们能再接再厉。

— 某教师语录

对于一个长度为 m 的整数序列 $c = [c_1, c_2, \cdots, c_m]$，
我们定义它的**均值** $f(c)$ 为序列元素的算术平均值。即：

$$f(c) = \frac{1}{m} \sum_{i=1}^{m} c_i$$

我们定义它的**达标率** $g(c)$ 为序列中**大于**均值的元素所占比例。即：

$$g(c) = \frac{1}{m} \sum_{\substack{1 \le i \le m \\ c_i > f(c)}} 1$$

给定一个长度为 n 的正整数序列 $[a_1, a_2, a_3, \cdots, a_n]$。请你对于 $k = 2, 3, 4, \cdots, n$ 依次计算：
如果从它全部的 $C(n, k)$ 个长度为 k 的子序列中**等概率随机**选取一个子序列，求其达标率的期望值。
由于答案可能为分数，请输出其对 998 244 353 取模的结果。
具体而言，可以证明，答案可以被表示为最简分数 $\frac{p}{q}$（p 与 q 互质且 q 不被 998 244 353 整除）。你需要输出满足 $0 \le x < 998\,244\,353$ 且 $x \cdot q \equiv p \pmod{998\,244\,353}$ 的整数 x。

Input

第一行一个整数 n（$2 \le n \le 50$），表示序列长度。
第二行 n 个整数 $a_1, a_2, a_3, \cdots, a_n$（$0 \le a_i \le n$）表示序列中的元素。

Output

输出 $n - 1$ 个整数，依次表示 $k = 2, 3, 4, \cdots, n$ 时的答案对 998 244 353 取模后的结果。

Examples

standard input
6 1 1 4 5 1 4
standard output
765320671 66549624 499122177 499122177 499122177

standard input
9 9 9 8 2 4 4 3 5 3
standard output
207967574 122799901 485257672 179050178 297096534 194103069 623902721 332748118

Problem I. 简单的数字运算 (II)

时间限制: 3 seconds
空间限制: 512 megabytes

定义函数 $F(x)$ 如下（其中 x 为正整数）：
若 x 的质因数分解结果为：

$$x = \prod_{i=1}^{k} p_i^{\alpha_i}$$

则：

$$F(x) = \prod_{i=1}^{k} \alpha_i^{p_i}$$

其中 p_i 是质数，α_i 是正整数。
特别的，我们规定 $F(1) = 1$。
给定一个正整数 N，请你计算以下表达式的值：

$$S(N) = \sum_{i=1}^{N} F(i)$$

由于结果可能很大，请输出 $S(N)$ 对 $998\,244\,353$ 取模的结果。

Input

仅一行，包含一个正整数 N（$1 \le N \le 10^{11}$）。

Output

输出一行，包含一个整数，表示 $S(N)$ 对 $998\,244\,353$ 取模的结果。

Examples

standard input	standard output
10	28
114514	726101213

Problem J. Ever Forever (II)

时间限制: 8 seconds
空间限制: 512 megabytes

无情的大火吞噬了整个音羽城，在教堂高耸尖塔的阴影下，瑞希的人生轨迹与乐观善良的优子交织在了一起。日子一天天过去，优子身上穷且益坚的意志，和对未来坚定不移的憧憬，为瑞希燃起了久违的温暖。

时光流转，季节更迭，岁月渐渐模糊成遥远的记忆。瑞希发现自己又一次站在了教堂前那象征命运的十字路口。她用微微颤抖的手，温柔地放下了一束鲜花，花瓣仿佛在轻声诉说着早已逝去的往昔。优子曾说过的一句句话，在瑞希的脑海中回响，那是时光也无法抹去的旋律。“即便最残酷的命运敲响了门，也要始终带着微笑去面对。” 这些话早已成为了瑞希的座右铭，引领着她度过在孤儿院一个个孤寂的夜晚。

瑞希的唇角勾起一抹灿烂的微笑，这笑容仿佛承载着千万个回忆的重量。泪水却在眼眶中打转，随时都可能夺眶而出，无声地证明着，痛苦与思念仍在她心中萦绕不去。那一刻，在过去与现在的交汇中，瑞希仿佛又变回了那个孤苦伶仃的小女孩，在教堂前，哭泣着奔跑着，去拥抱那个支撑自己踟蹰至今的优子姐姐。

瑞希有一个长度为 n 的只包含小写字母的字符串 s，她想要知道每一对 `e` 与 `f` 的距离之和，其中 `e` 出现在 `f` 之前。形式化的，她想要计算下列和式：

$$\mathrm{dis} = \sum_{\substack{1 \le i < j \le n \\ s_i = \mathtt{e} \\ s_j = \mathtt{f}}} (j - i)$$

瑞希接下来会按照顺序对字符串 s 进行 m 次修改操作，第 i 次修改操作会将 s_{p_i} 修改为 c_i。

需要注意，这里的修改是永久的。也就是说，每次修改会直接改变字符串，后续操作均基于前一次修改后的结果。

请你帮瑞希计算一下初始状态下字符串的 dis 值以及每次修改操作后字符串的 dis 值。

Input

第一行包含两个整数 n 和 m（$1 \le n, m \le 5 \cdot 10^5$）。

第二行包含一个长度为 n 的字符串 s，s 只包含小写字母。

接下来 m 行，第 i 行包含一个整数 p_i（$1 \le p_i \le n$）和一个小写字母 c_i，表示一次修改操作。

Output

输出 $m+1$ 行，每行包含一个整数，分别表示初始状态下，以及第 $1, 2, 3, \cdots, m$ 次操作后，字符串的 dis 值。

Example

standard input	standard output
12 3 efefyuukoefe 5 e 6 f 3 x	24 30 39 27