

河南省第十七届 ICPC 大学生程序设计竞赛
The 2026 ICPC China Henan Provincial Programming
Contest

现场赛
Onsite Round



题目列表
Problem List

A	来点离线做法
B	停车难题
C	数学的大厦崩塌了
D	贪玩迷宫
E	赚的越多，赚的越少
F	阶乘的和
G	球
H	关注火花花喵
I	收益
J	神经网络图染色
K	取石子
L	最小生成图
M	蜗牛养殖

请勿在比赛开始前翻阅试题!
Do not open before the contest has started.

2026 年 5 月 24 日

Problem A. 来点离线做法

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

给你一个长为 n 的非负整数序列 a_1, a_2, \dots, a_n 。

接下来有 q 次询问，每次询问给定三个整数 l, r, k ，请你计算：

$$\sum_{i=l}^r \lfloor \frac{a_i}{k} \rfloor = \lfloor \frac{a_l}{k} \rfloor + \lfloor \frac{a_{l+1}}{k} \rfloor + \dots + \lfloor \frac{a_r}{k} \rfloor$$

Input

第一行输入两个正整数 n 和 q ($1 \leq n, q \leq 10^5$)，分别代表序列的长度和询问次数。

第二行包含 n 个非负整数 a_i ($0 \leq a_i \leq 10^5$)，表示序列里的每个数字。

接下来 q 行，每行包含三个整数 l, r, k ($1 \leq l \leq r \leq n, 1 \leq k \leq 10^5$)，表示每次询问。

Output

对于每个询问，输出一行一个非负整数，表示所求的和。

Example

standard input	standard output
10 5	2
7 6 5 4 5 2 1 7 2 3	1
4 9 5	2
5 6 3	0
1 2 6	10
2 10 9	
6 8 1	

Explanation

对于第一次询问 $l = 4, r = 9, k = 5$ 时，答案为：

$$\lfloor \frac{4}{5} \rfloor + \lfloor \frac{5}{5} \rfloor + \lfloor \frac{2}{5} \rfloor + \lfloor \frac{1}{5} \rfloor + \lfloor \frac{7}{5} \rfloor + \lfloor \frac{2}{5} \rfloor = 0 + 1 + 0 + 0 + 1 + 0 = 2$$

Problem B. 停车难题

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 的工作单位有一个无穷大的停车场，车位的编号从 0 开始依次递增：0, 1, 2, 3...

小 C 有一个癖好：每次开车上班时都要停在**编号最小的空车位**上。他认为这样可以积攒运气，让一天的工作都顺顺利利。

今天早上小 C 又开车来到了停车场，在手机上，app 告诉了他所有已被占用的车位编号。

请你根据 app 提供的信息，帮小 C 找出他今天应该停在几号车位。

Input

第一行输入一个整数 n ($1 \leq n \leq 10^3$) 表示已被占用的车位数量。

第二行包含 n 个非负整数 $a_1, a_2, a_3 \dots a_n$ ($0 \leq a_i \leq 10^9$) 表示已被占用的车位编号，保证 a_i 严格单调递增。

Output

输出一行一个整数，表示小 C 应该停的车位编号。

Example

standard input	standard output
5 0 1 2 4 5	3
5 1 2 3 4 5	0

Explanation

对于第一个样例，0, 1, 2 号车位已被占用，3 号未被占用，因此**编号最小的空车位**为 3，输出 3。

对于第二个样例，0 号车位未被占用，他就是**编号最小的空车位**，输出 0。

Problem C. 数学的大厦崩塌了

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 正在学分数约分，他吃惊地发现，计算 $\frac{16}{64}$ 时，将分子和分母中同时出现的数字 6 “删去”，竟然得到了正确的约分结果 $\frac{1}{4}$ 。他又随手试了几个，发现类似的例子都是正确的：

$$\begin{array}{r} \frac{16}{64} = \frac{\cancel{1}6}{\cancel{6}4} = \frac{1}{4} \\ \frac{19}{95} = \frac{\cancel{1}9}{\cancel{9}5} = \frac{1}{5} \\ \frac{138}{184} = \frac{\cancel{1}3\cancel{8}}{\cancel{1}8\cancel{4}} = \frac{3}{4} \end{array}$$

图 1

小 C 感觉数学的大厦崩塌了，他想知道在给定范围内，还有多少个这样的分数可以让数学的大厦崩塌。

形式化地说，对于给定的正整数 n ，请你计算有多少个四元组 (A, B, a, b) 满足以下**全部**条件：

- $1 \leq A < B \leq n$ ，即约分前要求分母在 n 范围内且 $\frac{A}{B}$ 是一个真分数；
- $1 \leq a < b$ ，即约分后 $\frac{a}{b}$ 是一个真分数；
- A, B, a, b 均不含前导零；
- $A \neq a$ ，即不能什么都不删；
- $A * b = B * a$ ，即删减后分数的值与原分数相等；
- a 和 b 没有公共数字，即删减后的分数不可继续删减；
- 对于每个数字 $d \in \{0, 1, \dots, 9\}$ ， d 在 A 中的出现次数 = d 在 B 中的出现次数 = d 在 a 中的出现次数 = d 在 b 中的出现次数，即分子分母删去的数字完全相同；
- 若将 A, B, a, b 视为字符串，满足 a 是 A 的子序列， b 是 B 的子序列；

Input

一个正整数 n ($1 \leq n \leq 10000$), 表示 B 的取值范围上界。

Output

输出一行一个整数, 表示满足条件的四元组 (A, B, a, b) 的数量。

Example

standard input	standard output
40	6
100	48
10000	218429

Explanation

对于第一个样例, 满足条件的六个四元组分别是:

$$\frac{10}{20} = \frac{1}{2} \quad \frac{10}{30} = \frac{1}{3} \quad \frac{20}{30} = \frac{2}{3} \quad \frac{10}{40} = \frac{1}{4} \quad \frac{20}{40} = \frac{2}{4} \quad \frac{30}{40} = \frac{3}{4}$$

对于第二个样例, 除了 A 和 B 均为 10 的倍数的情况, 有四种情况比较特殊:

$$\frac{16}{64} = \frac{1}{4} \quad \frac{26}{65} = \frac{2}{5} \quad \frac{19}{95} = \frac{1}{5} \quad \frac{49}{98} = \frac{4}{8}$$

请注意, 同一个 $\frac{A}{B}$ 可能通过不同的删减方式得到不同的合法四元组。例如 $\frac{3163}{6326}$, 可以删减为 $\frac{13}{26}$ 或 $\frac{31}{62}$, 因此四元组 $(3163, 6326, 13, 26)$ 和 $(3163, 6326, 31, 62)$ 要分别计数。

以下是一些不合法的例子和原因:

- $\frac{604}{6040}$: 虽然可以通过删减得到 $\frac{4}{40}$ 且 $\frac{604}{6040} = \frac{4}{40}$, 但是 $\frac{4}{40}$ 有公共数字 4, 仍可继续删减, 不满足条件。
- $\frac{597}{995}$: 虽然可以通过删减得到 $\frac{57}{95}$ 且 $\frac{597}{995} = \frac{57}{95}$, 但是 $\frac{57}{95}$ 有公共数字 5, 仍可继续删减, 不满足条件。
- $\frac{893}{940}$: 虽然 $\frac{893}{940} = \frac{38}{40}$, 但将 38 视为字符串, 它不是 893 的子序列, 不满足条件。

Problem D. 贪玩迷宫

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 正被困在一个 n 行 m 列的二维迷宫里，这个迷宫有若干个终点（用大写字母'O'表示），到达这些终点位置就可以逃出迷宫。在非终点的位置有一个导向标，写着 UDLR 四个字母中的一种，在这些位置上，必须按 UDLR 的指示向上、下、左、右移动。

小 C 可以任选一个**非终点**位置出发，可能被困死在循环里，可能成功到达终点中的一个，还有可能撞上边界。小 C 有一次作弊操作，可以选择在出发前修改一个非终点位置的导向标。同时小 C 也比较贪玩，他想要在保证**能到达终点**的前提下尽可能多地在迷宫里移动。

如果不使用作弊操作，可能每个**非终点**位置都无法到达终点。但是通过使用作弊操作可以保证存在至少一个能到达终点的**非终点**位置。请问至多使用一次作弊操作后，保证**能到达终点**的前提下他最多可以移动几次？

Input

第一行一个正整数 T ($1 \leq T \leq 10$) 表示一共有 T 个测试用例。每个测试用例占 $n + 1$ 行，第一行两个整数 n, m ($2 \leq n, m \leq 1000, \sum n * m \leq 10^6$) 表示二维迷宫的行和列。

接下来 n 行，每行一个长为 m 的字符串，表示二维迷宫的每一行。保证字符串里只包含'O','U','D','L','R'，分别表示终点，向上，向下，向左，向右，至少存在一个终点位置，至少存在一个非终点位置。

Output

对于每个测试用例输出一行一个整数，代表答案。

Example

standard input	standard output
3	8
3 3	20
DLL	6
DOU	
RRU	
3 7	
RDLLLLL	
UDRRROU	
URRRRRU	
3 3	
ODL	
ROL	
RRU	

Explanation

对于第一个地图，如果不做修改，小 C 从任意一个**非终点**位置出发都会被困死在循环里。如果小 C 将第一行第二列位置的'L' 修改为'D'，再从第一行第一列位置出发就可以移动 8 次。还有其他的可行修改，能到达终点的前提下，能移动的次数一定小于等于 8。

对于第二个地图，将第一行第三列的'L' 修改为'D'，再从第三行第一列位置出发可以移动 20 次。

对于第三个地图，将第二行第三列的'L' 修改为'U'，再从第三行第一列位置出发可以移动 6 次。

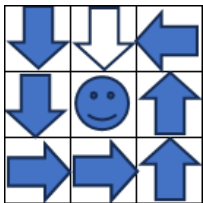


图 2

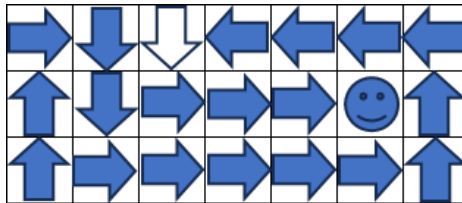


图 3

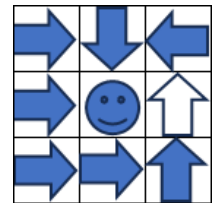


图 4

Problem E. 赚的越多，赚的越少

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 正在岔分宇宙里探索（和某个米哈游游戏玩法无关），最初他余额为 0。设小 C 当前余额为 x ，在每个环节中会依次进行以下两个操作：

1. 获得 X 元，余额变为 $x + X$ 元（鲁珀特帝国机械齿轮）；
2. 损失 $\lfloor \frac{x+X}{K} \rfloor$ 元（永动咕咕钟）；

即每个环节结束后余额变为 $x + X - \lfloor \frac{x+X}{K} \rfloor$ 元。

他赚的越多，损失的越多；损失的越多，赚的越少。因此，他赚的越多，赚的越少。

岔分宇宙里有 n 个环节，他想让你帮他算算 n 个环节之后，他的余额是多少？

Input

一行输入三个由空格隔开的正整数 n, X, K ($1 \leq n \leq 10^{18}$, $1 \leq X \leq 1000$, $1 \leq K \leq 1000$) 表示环节数量，每个环节获得的钱数和损失计算中的除数。

Output

输出一行一个非负整数，表示 n 个环节后小 C 的余额是多少。

Example

standard input	standard output
100000000 10 1	0
3 1000 2	875

Explanation

对于第一个样例，由于 $K = 1$ ，每个环节损失 $\lfloor \frac{x}{1} \rfloor$ 元，恰好将获得的 10 元全部扣光，因此每个环节后余额均为 0。

对于第二个样例：

- 第 1 个环节后余额为 $0 + 1000 - \lfloor \frac{1000}{2} \rfloor = 500$ ；
- 第 2 个环节后余额为 $500 + 1000 - \lfloor \frac{1500}{2} \rfloor = 750$ ；
- 第 3 个环节后余额为 $750 + 1000 - \lfloor \frac{1750}{2} \rfloor = 875$ ；

因此 3 个环节后余额为 875。

Problem F. 阶乘的和

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 刚学习了阶乘的定义: $n! = 1 \times 2 \times 3 \times \cdots \times n$ 。他发现这个数字增长极快, 于是想算算 $1 \sim n$ 的所有阶乘之和, 也就是 $S = \sum_{i=1}^n i!$ 。

这个和太大了, 所以请你帮他求出 $S \bmod 10000$ 的值。

Input

第一行输入一个正整数 n ($1 \leq n \leq 10^9$) 表示求和的项数。

Output

输出一行一个非负整数, 表示 $S \bmod 10000$ 的值。

Example

standard input	standard output
3	9
10	7913
20	313

Explanation

$n = 3$ 时, $S = \sum_{i=1}^3 i! = 1! + 2! + 3! = 1 + 2 + 6 = 9$, $S \bmod 10000 = 9$ 。

$n = 10$ 时, $S = \sum_{i=1}^{10} i! = 1 + 2 + 6 + 24 + 120 + 720 + 5040 + 40320 + 362880 + 3628800 = 4037913$,
 $S \bmod 10000 = 7913$ 。

Problem G. 球

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

在一个空间直角坐标系中有两个球，第一个球的球心在原点上，给定第二个球的球心和两个球的半径，求这两个球重叠部分的体积。

Input

第一行输入一个整数 T ($1 \leq T \leq 10^5$)。

接下来 T 行，每行五个空格隔开的整数 x, y, z, r_1, r_2 。 (x, y, z) 表示第二个球心的坐标， r_1 表示第一个球的半径， r_2 表示第二个球的半径 ($0 \leq x, y, z, r_1, r_2 \leq 10^4$)。

Output

对于每次询问，请输出一行一个实数表示两个球重叠部分的体积，当你的输出与标准答案之间的绝对误差或相对误差小于 10^{-6} 时视作正确。

Example

standard input	standard output
4	107.741635
10 10 10 10 10	4188.790205
0 0 10 30 10	16479.270375
0 10 10 20 20	0.000000
20 20 20 10 10	

Explanation

对于第一，三组数据两球相交，第二组数据第二个球被第一个球完全包含，第四组数据两球不相交。

Problem H. 关注火花花喵

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 最近沉迷某个二次元回合制游戏，游戏中有个叫做火花角色，小 C 格外喜欢她的类似抽奖的技能机制。经过小 C 的大胆猜想，火花的技能大概是这样的：从一个初始包含 3 个“红红火火”球和 17 个“恍恍惚惚”球的奖池中等概率不放回地抽取小球，总共 20 个球。每次抽奖遵循以下规则：

- 放一次技能期间最多进行 20 次抽奖，放完技能奖池重置。
- 抽到“红红火火”球：获得 2 点笑点和 2 点战技点。
- 抽到“恍恍惚惚”球：获得 1 点笑点。
- 整个过程中最多只能抽出 3 次“红红火火”球，因为只有 3 个“红红火火”球。
- 前三发抽奖中至少有一次抽出“红红火火”球。
- 如果连续六次抽到“恍恍惚惚”球，则下一次抽奖必定是“红红火火”球，已经抽完了所有的“红红火火”球时此规则失效。

此外，你还拥有三种战斗资源：爆点、战技点和战技点上限，每次抽奖需要使用一个爆点或者一个战技点。特殊的，如果同时拥有爆点和战技点则优先使用爆点，任何时刻战技点数量都不能超过战技点上限，增加战技点时，超出战技点上限的部分会消失。

小 C 某场游戏会使用火花放 T 次技能，给定每次放技能时的爆点数、战技点数和战技点上限，分别用 n, m, k 表示，假设每次放技能小 C 会尽可能多的抽奖，求每次放技能获得的笑点的期望值。

Input

第一行输入一个整数 T ($1 \leq T \leq 10^4$)。

接下来 T 行，每行三个空格隔开的整数 n, m, k 分别表示爆点数、战技点数和战技点上限 ($m \leq k$, $0 \leq n, m, k \leq 15$)。

Output

对于每次询问，请输出一行一个数表示此次技能获得的笑点的期望值，结果四舍五入保留到整数。

Example

standard input	standard output
3	2
0 1 5	11
3 4 5	23
8 8 10	

Problem I. 收益

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

给定一棵以 1 号节点为根的树，树上有 n 个节点，每个节点 i 有一个价值为 a_i 的物品，每个节点只有一个物品。树上有 m 个人，第 i 个人初始位于 p_i 节点，不同的人可以在同一个节点。每个人可以沿着树边向深度更大的方向移动，每到达一个节点，他可以选择拿走该节点上的物品，每个物品只能被拿走一次。每个人可以移动任意步，也可以选择不动。

所有人按照最优策略行动，求最大能拿到的物品总价值。

Input

第一行输入两个整数 n, m ，分别表示树上点的个数和人的个数 ($1 \leq n, m \leq 3 \times 10^5$)。

接下来 $n - 1$ 行，每行两个用空格隔开的整数 u, v ，表示 u, v 之间存在一条无向边 ($1 \leq u, v \leq n$)。

再接下来一行有 n 个用空格隔开的整数 $a_1, a_2, a_3 \dots a_n$ ， a_i 表示第 i 个节点物品的价值 ($0 \leq a_i \leq 10^9$)。

再接下来一行有 m 个用空格隔开的整数 $p_1, p_2, p_3 \dots p_m$ ， p_i 表示第 i 个人所在节点的编号 ($1 \leq p_i \leq n$)。

Output

请输出一行一个数表示最大能拿到的物品总价值。

Example

standard input	standard output
6 2 1 6 5 6 6 4 1 2 2 3 2 4 6 3 1 2 1 2	17

Explanation

第一个人从 1 号节点走到 4 号节点，第二个人从 2 号节点走到 3 号节点，每个人走到每个点都拿走所在节点的物品时获得最大总价值 17。

Problem J. 神经网络图染色

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

神经网络结构是模仿生物神经系统构建的计算模型，由输入层、隐藏层和输出层构成，通过调整神经元连接权重实现信息处理，在人工智能、自动控制等领域具有广泛应用。——百度百科

小 C 正在研究一个具有 n 层隐藏层的神经网络图，可以把它视为一个无向图。节点编号如下：

- 输入层：0 号节点；
- 第 k 层隐藏层 ($1 \leq k \leq n$)： $3k - 2$ 、 $3k - 1$ 、 $3k$ 号节点；
- 输出层： $3n + 1$ 号节点；

连边规则如下：

1. 输入层的 0 号点与第 1 层隐藏层的 3 个节点 (1, 2, 3) 均有连边。
2. 相邻隐藏层之间，第 k 层隐藏层和第 $k + 1$ 层隐藏层连边规则如下： $3k - 2$ 和 $3k + 1, 3k + 2$ 有连边， $3k - 1$ 和 $3k + 1, 3k + 3$ 有连边， $3k$ 和 $3k + 2, 3k + 3$ 有连边。
3. 第 n 层隐藏层的 3 个节点 ($3n - 2, 3n - 1, 3n$) 与输出层的 $3n + 1$ 号点均有连边。

例如当 $n = 4$ 时，图的结构如下：

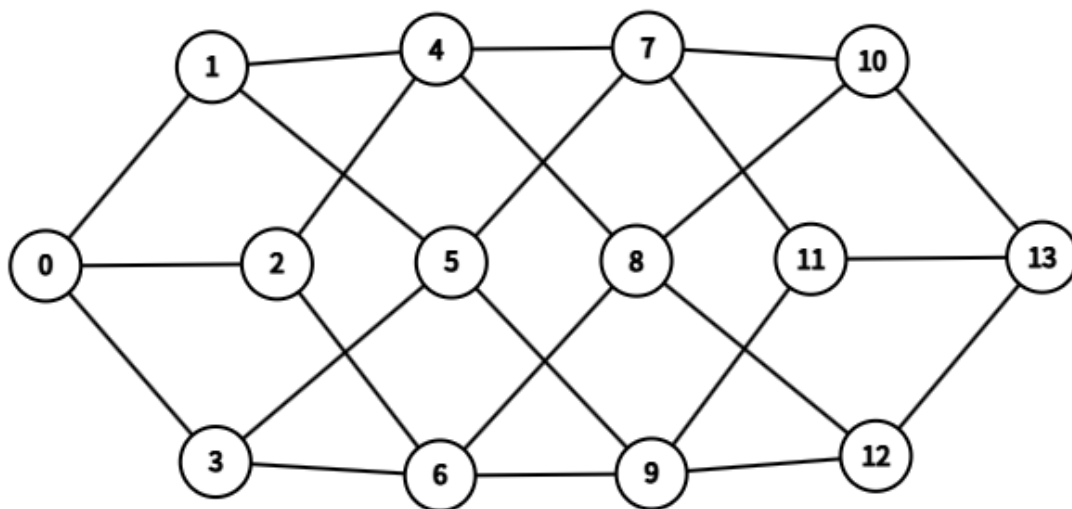


图 5

小 C 打算用 m 种颜色给图中所有节点染色，要求有连边的两个节点颜色不同，求染色方案数对 998244353 取模的结果。

Input

输入一行两个空格隔开的正整数 n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 1000$), 表示隐藏层的层数和可用颜色的数量。

Output

输出一行一个非负整数, 表示染色方案数对 998244353 取模的结果。

Example

standard input	standard output
1 3	30
4 2	2
2 10	28478970
100000 1000	937751774

Explanation

对于样例 1, 图像如下所示:

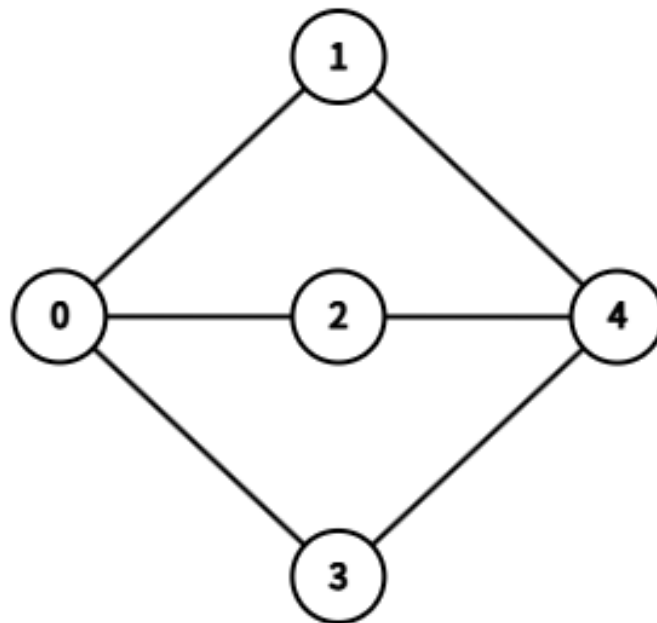


图 6

1, 2, 3 三个节点颜色相同的方案有 12 种, 三个节点恰好使用两种颜色的方案有 18 种, 不可能有三种颜色。

对于样例 2, 图像见题干。合法方案有 2 种。

Problem K. 取石子

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

桌子上有 n 堆石头，第 i 堆石头里有 a_i 个石头。Alice 和 Bob 轮流操作，Alice 先手。每次操作时，当前操作的玩家需要选择至少一个石子数为偶数的堆，将选中的每一堆均分为两堆。

若所有堆的石子数均为奇数，即当前玩家无法进行任何操作，则当前玩家判负。

两个人都足够聪明，采取最优策略，请判断谁会获胜。

Input

输入一共两行。

第一行一个整数 n 表示石头有几堆，保证 $1 \leq n \leq 10^3$ 。

第二行 n 个空格隔开的整数 a_i 表示每堆石头的数量，保证 $1 \leq a_i \leq 10^5$ 。

Output

如果 Alice 会赢，请输出 Alice，否则输出 Bob。

Example

standard input	standard output
1 3	Bob
2 2 4	Alice

Explanation

对于第一个样例，由于 Alice 先手无法操作，所以 Bob 获胜。

对于第二个样例，Alice 可以将石子变为 1 1 4，此时 Bob 只能变为 1 1 2 2，Alice 再操作为 1 1 1 1 1 1 后即可获胜。

Problem L. 最小生成图

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 最近在研究最小生成树，他灵机一动想出这么一道题：

给定一个 n 个点 m 条边的无向连通图，第 i 条边连接 u_i 号节点和 v_i 号节点，边权为 w_i ，图中可能存在重边和自环。

小 C 从中取出任意多条边并且以任意顺序将边排列，即构造一个序列 $a_1, a_2, a_3, \dots, a_k$ ($n-1 \leq k \leq m$, $1 \leq a_i \leq m$ 且 a_i 互不相同)，进行 k 次操作，第 i 次操作选择第 a_i 条边并连接该边对应的两个点 u_{a_i} 和 v_{a_i} ，此时这条边的代价为 $\frac{w_{a_i}}{i}$ ，总代价为选中的每条边代价的和，求能使 n 个点联通的所有方案中总代价的最小值。

Input

第一行输入两个整数 n, m ($1 \leq n \leq 3 \times 10^3, n-1 \leq m \leq 5 \times 10^3$)。

接下来 m 行，每行三个空格隔开的整数 u_i, v_i, w_i ，表示第 i 条边连接 u_i 号节点和 v_i 号节点且边权为 w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9$)。

Output

请输出一行一个数表示总代价的最小值，结果四舍五入到整数。

Example

standard input	standard output
5 9 1 2 1 2 3 2 1 3 3 2 3 4 1 3 5 1 3 50 1 4 60 1 5 70 4 5 80	25

Explanation

依次选择第 1, 2, 3, 4, 5, 7, 8 条边，此时总代价最小，为 $\frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{4}{4} + \frac{5}{5} + \frac{60}{6} + \frac{70}{7} = 25$

Problem M. 蜗牛养殖

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

小 C 正在 n 个盒子和 $n - 1$ 个双向通道里养蜗牛，盒子标记为 $1 \sim n$ ，任意两个盒子通过通道可以互通互联。其中有 k 个盒子里现在有蜗牛，分别为 x_1, x_2, \dots, x_k 号盒子。为了避免蜗牛“串门”，小 C 想要给每个双向通道关闭一个方向，修改为单向通道，使得养有蜗牛的 k 个盒子两两之间都不能互通互联。你能帮他算算有多少种合法的方案吗？

由于方案数可能非常大，请输出方案数对 998244353 取模的结果。

Input

输入第一行两个由空格隔开的整数 n, k ，表示盒子的数量和有蜗牛的盒子的数量，保证 $1 \leq n \leq 10^5$ ， $1 \leq k \leq \min(n, 4)$ 。

第二行输入 k 个空格隔开的整数 a_i 表示有蜗牛的盒子的编号，保证 $1 \leq a_i \leq n$ ， a_i 两两不同。

接下来的 $n - 1$ 行，每行两个空格隔开的整数 x, y 表示双向通道连接的盒子的编号，保证 $1 \leq x, y \leq n$ ， $x \neq y$ ，盒子与通道形成了一棵树。

Output

输出一行一个非负整数表示方案数对 998244353 取模的结果。

Example

standard input	standard output
5 1 2 1 2 3 1 2 4 2 5	16
5 2 3 4 1 2 3 1 2 4 2 5	12
5 3 3 4 5 1 2 3 1 2 4 2 5	6

Explanation

对于第一个样例，有五个盒子，其中 2 号盒子里有蜗牛，以 1 号点为根时树的形态如下图所示。由于只有一个盒子里有蜗牛，所以任意的方案都合法。

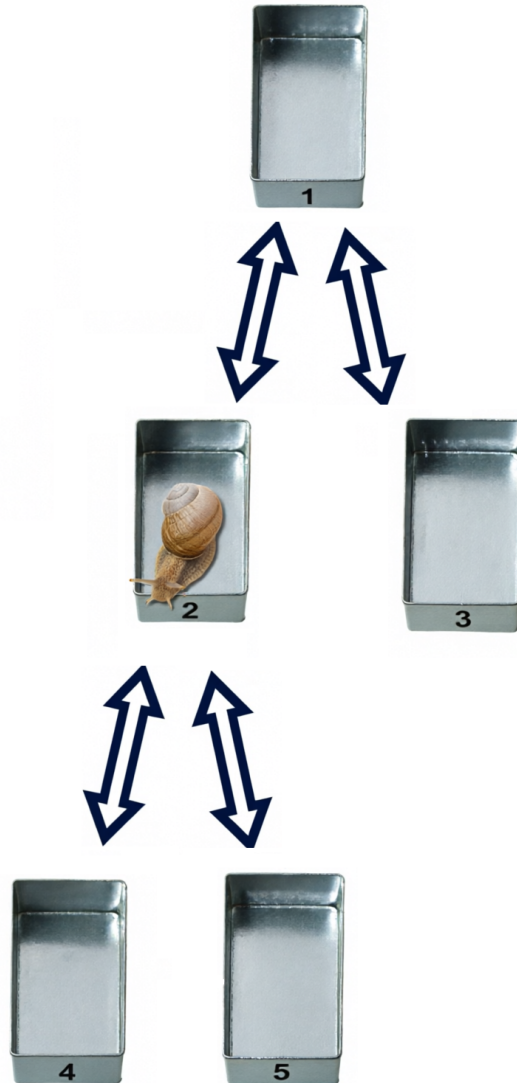


图 7

对于第二个样例，有五个盒子，树的形态和样例一一样，蜗牛所在位置改为了 3 号 4 号盒子里有蜗牛。不合法的方案有四个，因此答案是 12。