



## 第二十一届中国计量大学校ACM团队赛暨《ACM竞赛实训》期末考试

### 正式赛

2024 年 5 月 25 日

### 题目概况

题号	题目名	时间限制/ms	空间限制/KB
1	我是构造	1000	32268
2	方块世界の苹果收集装置	1000	32268
3	我要上CJLU!	1000	32268
4	Lca? Lca!	600	131072
5	A Simple Game	500	32268
6	方块世界の时光机	1000	32268
7	树上随机游走	1000	32268
8	我们都是小青呱，WA WA WA WA WA!	800	32268
9	Creeper and Cats	1000	32268
10	我也是背包问题	500	64536
11	方块世界の魔法小径 (easy version)	1000	32268
12	方块世界の魔法小径 (hard version)	5000	32268
13	第二名>_<	1000	32268



# 1: 我是构造

Time Limit: 1000MS

Memory Limit: 32268KB

### Description

给你一个**正偶数**  $x$  和两个正整数  $n, m$  , 请你判断是否可以构造一个  $n \times m$  的矩阵, 且满足:

- 1. 矩形中的每个元素都是非负整数
- 2. 矩阵的每一行、每一列的异或和都相等
- 3. 矩阵的每个元素的和为  $x$

如果可以构造则输出 **"yes"** , 否则输出 **"no"** (输出不包含引号)

关于异或运算: 如果两个非负整数  $a, b$  对应二进制位上的数字相同, 则该位的异或结果为 0 , 否则为 1 。例如, 12 与 9 的异或为  $12 \oplus 9 = (1100)_2 \oplus (1001)_2 = (0101)_2 = 5$ , 3 与 5 的异或为  $3 \oplus 5 = (011)_2 \oplus (101)_2 = (110)_2 = 6$

更多数的异或运算按照从左到右的顺序执行。例如,  $1 \oplus 2 \oplus 3 = (1 \oplus 2) \oplus 3 = 0$  。此外, 容易证明异或具有交换律和结合律, 即  $a \oplus b = b \oplus a$  ,  $a \oplus b \oplus c = (a \oplus b) \oplus c = a \oplus (b \oplus c)$  。因此, 和加法一样, 在计算多个数的异或和时, 计算的次序并不重要。

### Input

输入仅一行, 为三个数  $n, m, x(4 \leq n, m \leq 10^4, 2 \leq x \leq 100$  且  $x$  为偶数 ) 。

### Output

输出 'yes' 或 'no'(不带引号)。

### Sample

standard input	standard output
4 4 2	no
4 5 6	yes

### Hint

第二个样例的解释:

注意到矩阵  $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$  满足条件, 所以输出 yes

## 2: 方块世界の苹果收集装置

Time Limit: 1000MS

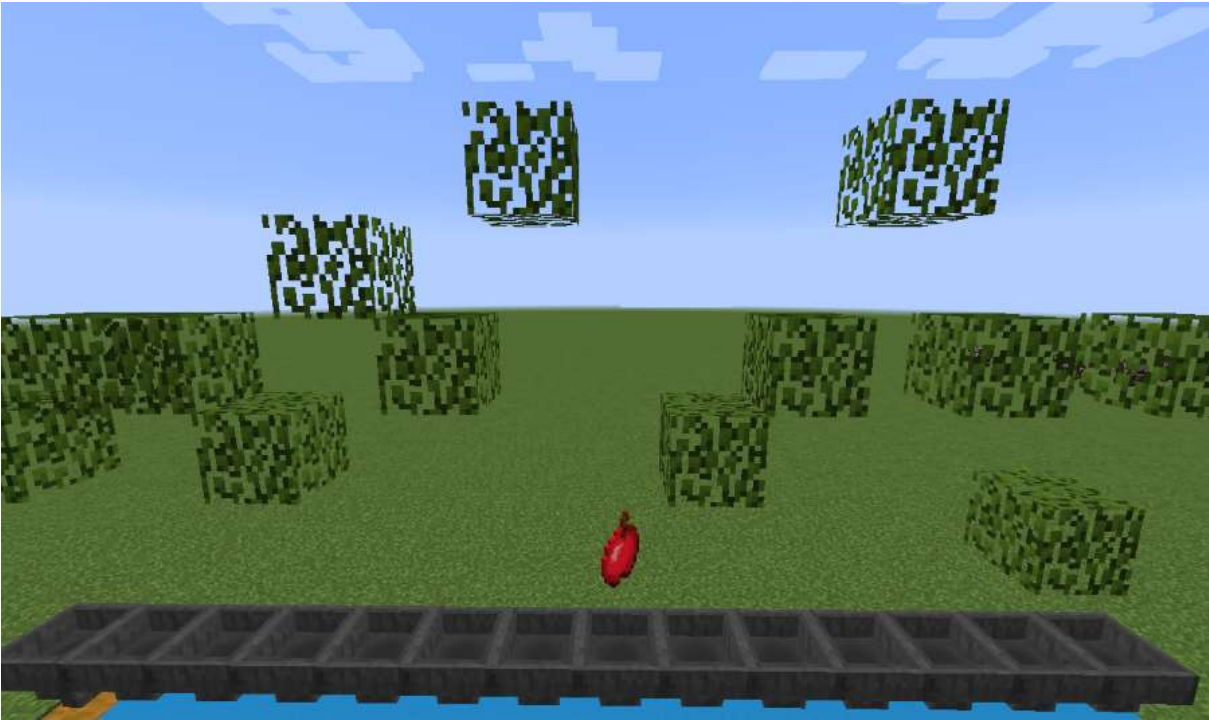
Memory Limit: 32268KB

### Description

*SevenGod*和*Taibo*乘坐时光机来到1.19.2之后见到了正在等待树叶消失掉落苹果(MC特性)的*ManInM00N*,为了让*ManInM00N*陪乘坐时光机而来的*SevenGod*和*Taibo*出门冒险,三人决定使用漏斗来等待树叶凋零后捡苹果,三人掏出了全部家底的铁锭,合成了一个长度为 $d$ 的漏斗。

本题假设所有树叶都在一条直线上,且漏斗的长度为 $d$ ,持续时间为 $w$ (时间一到漏斗就会消失)

聪明的大数学家通过一波完美的数学+自然规律推导得出了接下来 $n$ 个苹果的掉落时间 $t_i$ 和掉落位置 $x_i$ ,请你告诉三人他们能收集到的苹果的最大值 $res$ ,即存在一个放置位置 $L$ 和放置时间 $S$ ,使得漏斗在 $[S, S + W - 1]$ 的时间区间内能收集到 $res$ 个位于 $[L, L + d - 1]$ 坐标区间内的掉落苹果。



### Input

第一行输入三个数 $n, w, d$ ,表示后面要掉下来的苹果数量,漏斗的持续时间,漏斗的总长(漏斗是一整个整体,不可分割)

- $1 \leq n \leq 1e5$
- $1 \leq w \leq 2e5$
- $1 \leq d \leq 2e5$

接下来 $n$ 行,每行输入两个数 $t_i, x_i$ ,表示第 $i$ 个苹果的位置为 $x_i$ 将在第 $t_i$ 秒落地。

- $1 \leq t_i \leq 2e5$
- $1 \leq x_i \leq 2e5$

### Output

输出一个整数 $res$ ,表示最多可以收集到的苹果数量

Sample

standard input	standard output
8 4 3 1 1 3 4 6 4 5 2 4 2 4 3 5 5 7 3	5

Hint

如果选择了 $S = 3$ 和 $L = 2$ ，那么说明漏斗持续时间区间为 $[3, 6]$ ，覆盖区间为 $[2, 4]$ ，最终，三人一共会得到以下五个苹果：

- 时间 $T_2 = 3$ 时，落在坐标 $X_2 = 4$ 处的苹果
- 时间 $T_3 = 6$ 时，落在坐标 $X_3 = 4$ 处的苹果
- 时间 $T_4 = 5$ 时，落在坐标 $X_4 = 2$ 处的苹果
- 时间 $T_5 = 4$ 时，落在坐标 $X_5 = 2$ 处的苹果
- 时间 $T_6 = 4$ 时，落在坐标 $X_6 = 3$ 上的苹果

不可能得到6个或更多的苹果，所以打印 5。

# 3: 我要上CJLU!

Time Limit: 1000MS

Memory Limit: 32268KB

## Description

杨学长在考研中取得了非常高的分数，可以上任何一所大学，但是你询问过后知道了杨学长第一志愿**只可能填三个大学**，CJLU、PKU、THU，同时你得到了一张纸条，纸条上都是**随机大写字母**，聪明的你马上知道这个纸条如何破译，规则如下：

- 只要从这一串字母中按原顺序得到每个字母，**第一个出现完所有字母的大学**（CJLU、PKU、THU）就是杨学长的第一志愿按原顺序得到每个字母的含义如下：  
（例：CJLPPPPUKKKK 中你可以按CJLPPPPUKKKK的顺序得到CJLU，但是你得不到PKU，因为CJLPPPUKKKK，你最多只能得到PK）。
- 我们很容易发现这种情况，对于字符串CJLPKU,在取到最后的U时，我们可以得到的大学同时有CJLU和PKU，这时候我们需要输出CJLU，**因为杨学长内心的志愿倾向是CJLU > PKU > THU。**
- 如何这串字符中你无法解密这三个大学中的任意一个，请输出NO。

此处故事纯属虚构，也在这里祝杨学长在未来的考研中取得佳绩！

## Input

输入一个字符串S，题目保证全部字符均为大写英文单词

- $1 \leq |S| \leq 5e4$

## Output

输出杨学长的第一志愿

- 输出仅有四种结果，CJLU,PKU,THU,NO

## Sample

standard input	standard output
PIOKTHU	PKU
JCOPKPLUPU	PKU
LCJOKPU	NO

# 4: Lca? Lca!

Time Limit: 600MS

Memory Limit: 131072KB

## Description

Lococ发现了 ManInM00N 是导致第二名的罪魁祸首，他感到很生气🙄，惩罚他玩一个小游戏



Lococ 给了ManInM00N 一棵具有  $n$  个结点的树，结点的下标为  $1 \dots n$ ，其中包含  $l$  个叶子结点。树的根结点为 1 且 树的根结点不被视作叶子结点。他要求 ManInM00N 求出  $l$  个集合的大小，求第  $i$  个集合时，需要从  $l$  个叶子结点内选出  $i$  个结点  $u_1, u_2, \dots, u_i$  进行求解

- 这里定义一个集合  $S$ ，我们从  $l$  个叶子中选出  $k$  个叶子结点  $u_1, \dots, u_k$
- 那么这个集合  $S = \{ \text{lca}(u_i, u_j) \mid 1 \leq i, j \leq k \}$

此处  $\text{lca}(u, v)$  表示顶点  $u$  和  $v$  的最近公共祖先

你可以帮他求解出对于  $1 \dots l$  中的对于每一个  $i$ ，集合  $S$  的元素数量最多为多少吗

## Input

第一行为一个正整数  $n$  ( $2 \leq n \leq 10^5$ ) 表示结点数  
第二行为  $n - 1$  个正整数  $p_2, p_3, \dots, p_n$  ( $1 \leq p_i \leq n$ ) 表示  $p_i$  和  $i$  之间有一条边

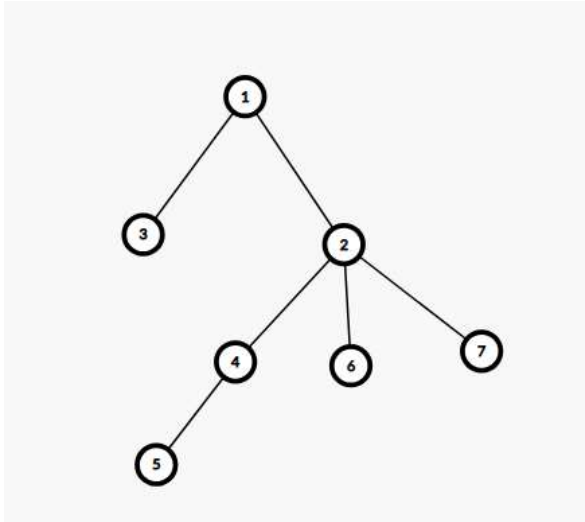
## Output

设  $l$  为树的叶子结点数量  
输出一行  $l$  个正整数，表示  $k = i$  时集合内最多可以有多少元素  
行末没有空格

## Sample

standard input	standard output
7 1 1 2 4 2 2	1 3 5 6

## Hint



在样例中，有4个叶子 $\{3, 5, 6, 7\}$

- $i = 1$ 时，选择任意一个叶子的答案都是 1
- $i = 2$  时，我们可以选择  $\{5, 7\}$  或者  $\{3, 7\}$ ，那么他们可以生成的集合为  $\{2, 5, 7\}$ 、 $\{1, 3, 7\}$
- $i = 3$  时，我们可以选择  $\{3, 5, 7\}$  或者  $\{3, 5, 6\}$ ，他们可以生成的集合为  $\{1, 2, 3, 5, 7\}$ 、 $\{1, 2, 3, 5, 6\}$  而  $\{5, 6, 7\}$  不行，因为他们可以生成的集合为  $\{2, 5, 6, 7\}$ ，元素数量为 4 比上述选法少。
- $i = 4$  时，我们选择全部的叶子，可以获得  $\{1, 2, 3, 5, 6, 7\}$

# 5: A Simple Game

Time Limit: 500MS

Memory Limit: 32268KB

## Description



作为 xxcdsg 的舔狗，Taibo 经常邀请 xxcdsg 一起玩无聊的游戏。

有一个长度为  $n$  的序列  $[a_1, a_2, \dots, a_n]$ ，Taibo 和 xxcdsg 轮流在这个序列中取一个非空子序列并删除该子序列直到序列为空，Taibo 是先手。游戏结束后，定义游戏的得分为 Taibo 和 xxcdsg 每次所取的子序列的异或和的总和。现在，Taibo 想要最大化这个得分，而 xxcdsg 则想要最小化该得分。如果双方都采取最佳策略的情况下，最终的得分将是多少？

关于非空子序列的定义：如果正整数  $i_1, i_2, \dots, i_k (k \geq 1)$  满足  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ ，那么  $[a_{i_1}, a_{i_2}, \dots, a_{i_k}]$  是  $[a_1, a_2, \dots, a_n]$  的一个非空子序列。例如， $[1, 2, 6]$ ,  $[1, 3, 5]$ ,  $[4]$ ,  $[1, 3, 5, 2, 4, 6]$  是  $[1, 3, 5, 2, 4, 6]$  的非空子序列，而  $[1, 2, 3]$ ,  $[4, 5]$ ,  $[1, 2, 3, 4, 5, 6]$  不是。

## Input

第一行为序列的长度  $n$ ，第二行为序列  $[a_1, a_2, \dots, a_n]$ 。

$$n \leq 2 \times 10^5, 0 \leq a_i < 2^{20}$$

## Output

输出一个数代表答案。

## Sample

standard input	standard output
1 1	1
2 3 1	4
3 1 1 1	1
4 1 1 1 1	2
3 5 3 2	10

## Hint

对于第二个样例：若 Taibo 选择  $[3, 1]$ ，则序列为空游戏结束，得分为  $3 \oplus 1 = 2$ ；若选择  $[3]$ ，而 xxcdsg 选择  $[1]$ ，则游戏得分为  $3 + 1 = 4$ 。故 Taibo 会选择第二个方案。

对于第五个样例：Taibo 会选择  $[5, 2]$ ，接着 xxcdsg 会选择  $[3]$ ，最终得分为  $(5 \oplus 2) + 3 = 10$



# 6: 方块世界の时光机

Time Limit: 1000MS

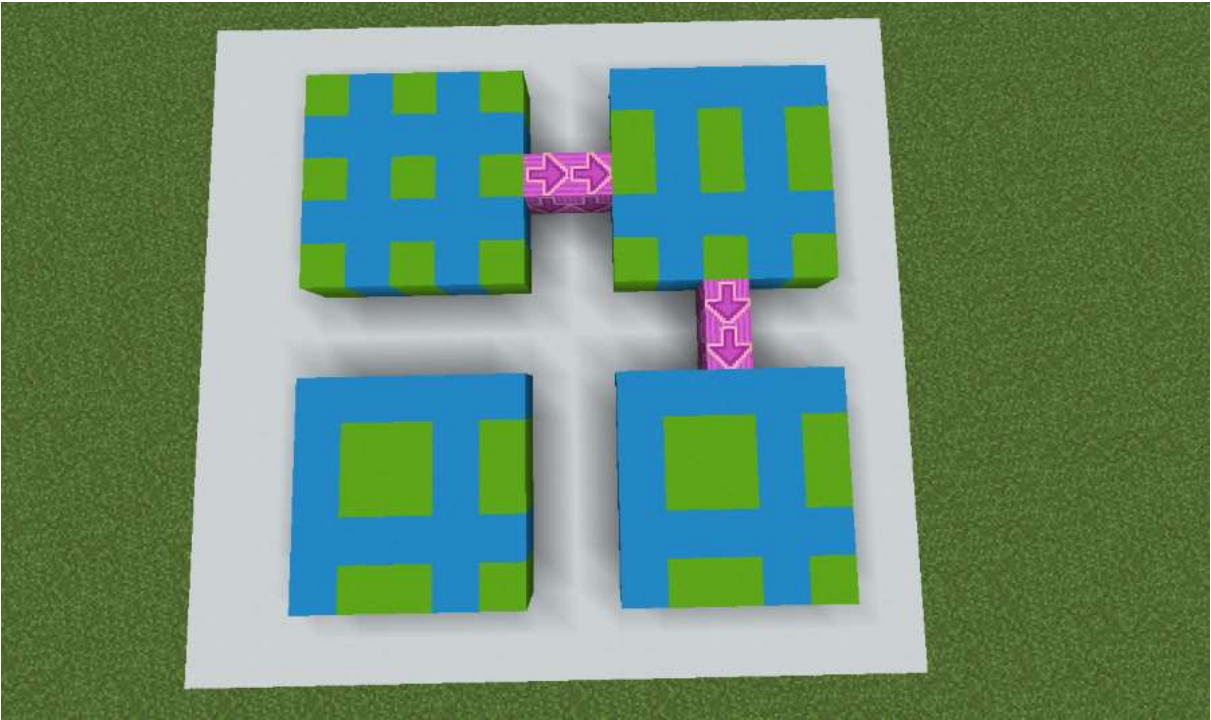
Memory Limit: 32268KB

## Description

众所周知, *Minecraft*的版本一直在不断更新, 但是方块世界的科学家们发明出来了一款时间机, 可以往返各个版本之间, 而这款时间机有一个启动装置, 启动装置分为两个  $n * m$  的矩阵, 分别为初始矩阵和目标矩阵, 只需通过一系列操作将初始矩阵还原为目标矩阵, 即可启动时间机。

根据科学家们立下的规则, 每次你能选择矩阵的第  $i$  行, 并将其与第  $i + 1$  行 ( $i < n$ ) 交换位置, 或者选择矩阵的第  $i$  列, 并将其与第  $i + 1$  列 ( $i < m$ ) 交换位置。

一天方块人 *SevenGod* 听说隔壁村的方块人 *ManInM00N* 在 1.19.2 happy play, *SevenGod* 决定拉上同村的 *Taibo* 去找 *ManInM00N* 游玩。但是 *SevenGod* 和 *Taibo* 实在是太懒了, 所以你能告诉他们最少需要几步能将初始矩阵还原为目标矩阵吗?



## Input

第一行输入两个整数  $n, m$

- $1 \leq n \leq 5$
- $1 \leq m \leq 5$

后面  $2 * n$  行每行  $m$  个数, 表示输入的初始矩阵  $A$  和目标矩阵  $B$

- $1 \leq A_{ij}, B_{ij} \leq 1e9$

## Output

输出一个整数  $x$ , 表示最少需要  $x$  步就可以将初始矩阵转换为目标矩阵

Sample

standard input	standard output
5 5 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 0 1 0 1 0 0 1 0 1 1 1 1 1 1 0 0 1 0	2
2 2 1 1 1 1 1 1 1 1000000000	-1
3 3 8 1 6 3 5 7 4 9 2 8 1 6 3 5 7 4 9 2	0

Hint

样例1的最小操作方案的其中一种具象化在题目的图中

其具体操作方案是

- 第一次选择第一行和第二行交换位置
- 第二次选择第一列和第二列交换位置

# 7: 树上随机游走

Time Limit: 1000MS

Memory Limit: 32268KB

## Description



众所周知，前段时间，Taibo 和 langgod 灵魂互换了。当时，Taibo 认真研究了如何在短时间内找到 langgod 的问题。

已知天性好动的 langgod 在一棵有  $n$  个结点的树上随机游走，树的结点从 1 开始编号。在 0 时刻，langgod 处在第  $r$  号结点上，且每过 1 秒，langgod 会随机向某个相邻的结点移动。 $t$  秒后，Taibo 赶到了 langgod 所在的树，他想知道此刻 langgod 可能在哪些结点上。

## Input

第一行有三个正整数  $n, r, t$

接下来有  $(n - 1)$  行，每行两个整数  $u, v$ ，代表结点  $u$  和结点  $v$  之间有一条边。

保证输入数据能够生成一棵树， $2 \leq n \leq 5 \times 10^4, 1 \leq r \leq n, 0 \leq t \leq 10^{18}$

## Output

第一行输出一个整数，代表 langgod 可能出现在几个结点上。

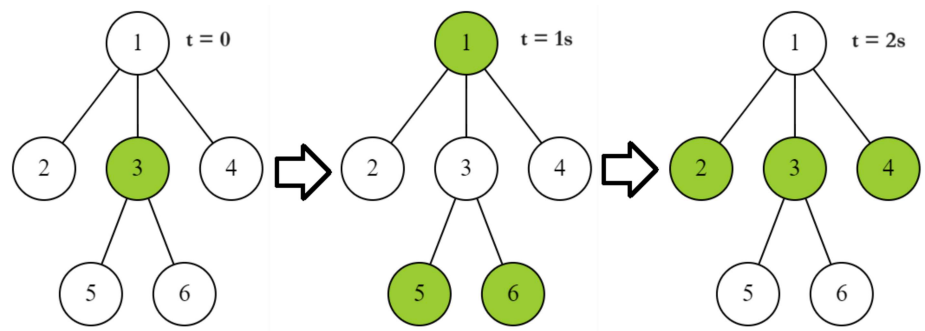
第二行按照递增顺序输出这些结点的编号。

## Sample

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

## Hint

对于样例，langgod 在不同时刻可能的位置如下图所示，其中绿色结点代表 langgod 此刻可能处在的结点：



# 8: 我们都是小青呱，WA WA WA WA WA!

Time Limit: 800MS

Memory Limit: 32268KB

Description



Taibo 和 Yearlidoggy 的小青蛙在某景区游玩。

这个景区有  $n$  个岛屿和  $m$  座双向的桥（均从 1 开始编号）。这些桥分为不同的种类。青蛙们不能免费使用这些桥。如果要使用某座桥，必须要购买对应的桥梁种类的会员卡，第  $i$  种桥对应的会员卡的价格为  $2^{i-1}$  个三叶草。拥有某种会员卡就可以通过指定种类的桥梁了。

现在，两只小青蛙想要从 1 号岛屿前往  $n$  号岛屿，但是都没有会员卡。请你告诉他们至少要花费多少三叶草才能抵达目的地。

## Input

第一行两个正整数  $n, m$

接下来  $m$  行，第  $i$  行有三个正整数  $u_i, v_i, w_i$  代表第  $i$  座桥连接岛屿  $u_i$  和  $v_i$ ，种类为  $w_i$ 。注意可能存在两个桥连接相同的两个岛屿， $u_i \neq v_i$

$2 \leq n \leq 10^4, 0 \leq m \leq 10^4, 1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10$

## Output

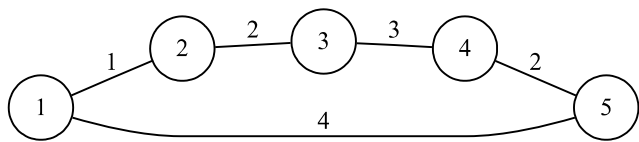
一个整数，表示至少需要花费多少三叶草。如果不能抵达目的地，输出 "-1"（不含引号）。

## Sample

standard input	standard output
5 5 1 2 1 2 3 2 3 4 3 4 5 2 1 5 4	7

Hint

对于样例，一共有以下两种办法抵达 5 号岛屿：



$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ，经过的桥的种类为 1, 2, 3，费用为  $2^{1-1} + 2^{2-1} + 2^{3-1} = 7$ 。

$1 \rightarrow 5$ ，经过的桥的种类为 4，费用为  $2^{4-1} = 8$ 。

综上，最小费用为 7。

# 9: Creeper and Cats

Time Limit: 1000MS

Memory Limit: 32268KB

## Description

在 Taibo 的某篇 MC 小说里，手无寸铁的苦力怕曾为了保护玩家，随手抱起一只旁边的猫作为自己的武器（猫猫冲锋枪）并冲锋陷阵.....



某天，Taibo 打开了尘封已久的 Minecraft，建造了  $n \times m$  个紧挨着的房间，这些房间构成了  $n \times m$  的网格。我们规定位于第  $x$  行第  $y$  列的房间的坐标为  $(x, y)$ 。其中，对于任意一个房间  $(i, j)$ ，如果  $i < n$ ，那么 Taibo 会放置一个单向铁门通往房间  $(i + 1, j)$ ；如果  $j < m$ ，那么 Taibo 会放置一个单向铁门通往房间  $(i, j + 1)$ 。

布置好这些房间之后，Taibo 突然想到了个好玩的游戏：

- Taibo 在房间  $(1, 1)$  放了一只苦力怕。这只苦力怕会不断地在自己所处的房间里面随机选择一个铁门并通过这个铁门到达新的房间（当然如果只有 1 个铁门的话就只能选择这个铁门），直到抵达房间  $(n, m)$
- Taibo 在其中某些房间（可能是 0 个房间或所有房间）各放了一只猫。这些猫分为人畜无害的小猫咪和炸弹猫
- 苦力怕很喜欢猫，如果在途中遇到某只小猫，苦力怕一定会带着这只小猫一起走。然而，一旦遇到炸弹猫，苦力怕在之前带上的所有小猫和这只炸弹猫都会消失（苦力怕不会消失）。

设随机变量  $X$  代表苦力怕抵达房间  $(n, m)$  后所携带的小猫的个数。求出随机变量  $X$  的期望值  $E(X)$ ，答案对  $10^9 + 7$  取模。

关于分数取模：可以证明答案一定可以化简为一个最简分数  $\frac{p}{q}$  的形式， $\frac{p}{q}$  模  $10^9 + 7$  的值即  $pq^{-1}$  模  $10^9 + 7$  的值，其值等于使得  $0 \leq x < 10^9 + 7$  且  $qx \equiv p \pmod{10^9 + 7}$  成立的整数  $x$  的值。

## Input

第一行一个正整数  $T$ ，代表有  $T$  组测试数据。

对于每个测试数据，第一行为两个正整数  $n, m$

接下来有  $n$  行长度为  $m$  的字符串，第  $i$  行的字符串的第  $j$  个字符代表房间  $(i, j)$  的状态，"." 代表房间里没有猫，"C" 代表房间里有一只猫，"B" 代表房间里有一只炸弹猫。

$T \leq 100, 1 \leq n, m \leq 1000$ ，各测试数据  $nm$  的总和不超过  $10^5$ 。

## Output

对于每组测试数据，输出所求答案。

Sample

standard input	standard output
3 2 2 .. .. 2 3 C.. ..C 3 3 .C. CBC .C.	0 2 500000005

Hint

对于第三个测试用例，下面列出苦力怕所有可能的移动轨迹：

$(1, 1) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (3, 2) \rightarrow (3, 3)$ ，概率为  $\frac{1}{4}$ ，最后苦力怕携带了 2 只猫。

$(1, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (3, 2) \rightarrow (3, 3)$ ，概率为  $\frac{1}{8}$ ，最后苦力怕携带了 1 只猫。

$(1, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (2, 3) \rightarrow (3, 3)$ ，概率为  $\frac{1}{8}$ ，最后苦力怕携带了 1 只猫。

$(1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow (3, 2) \rightarrow (3, 3)$ ，概率为  $\frac{1}{8}$ ，最后苦力怕携带了 1 只猫。

$(1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow (2, 3) \rightarrow (3, 3)$ ，概率为  $\frac{1}{8}$ ，最后苦力怕携带了 1 只猫。

$(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (3, 3)$ ，概率为  $\frac{1}{4}$ ，最后苦力怕携带了 2 只猫。

故  $E(X) = \frac{3}{2}$

# 10: 我也是背包问题

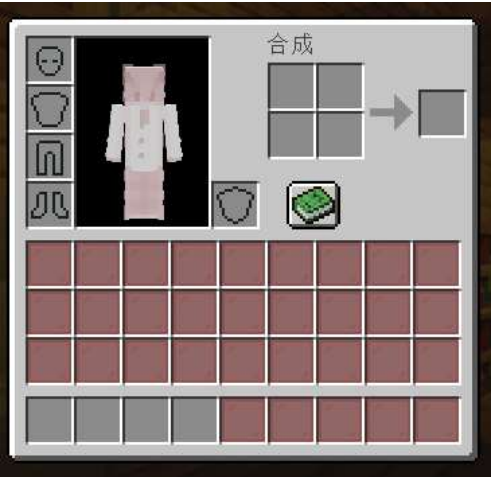
Time Limit: 500MS

Memory Limit: 64536KB

## Description

众所周知，玩*Minecraft*出门第一件事就是回家  
*ManInM00N*作为一只喜欢住山涧的仓鼠，在服务器的日常就是到处去朋友们家里打理，顺便捡点垃圾吃。

但是有一天，服务器出了一个bug，*ManInM00N*（的存档）没辣！，并且服务器内的机器也出现了问题，*ManInM00N*创了个新档去维修分布在  $n$  个地区的机器。  
由于是新存档，服务器的权限让*ManInM00N*有家不能回，故*ManInM00N*一开始的背包里面什么工具都没有，背包容量大小为  $k$ （也就是说可以放下  $k$  把工具，每一种工具的堆叠上限为 1）。



- 对于每个地区  $i$
- 需要用第  $a_i$  种工具维护，*ManInM00N*的背包内**必须**持有对应第  $a_i$  种的工具来进行维修。
  - 如果没有就要花费  $b_{a_i}$  个绿宝石的价格购买工具。
  - 如果在购买前背包是满的，他必须丢弃至少一把工具。

维护机器的路线是固定的，*ManInM00N* 必须按照  $1, 2 \dots n$  的顺序访问地区

*ManInM00N*的作风非常浪费，如果他丢掉了一个工具，那么这个工具他**不会**再捡回来使用（即使下一个地区可能要用到这个工具），即 *ManInM00N***随时可以扔掉背包内任意数量**的工具，并且这个操作是**不可逆**的。  
现在*ManInM00N*陷入了一个问题，怎样才能花**最少**的绿宝石维修好所有机器。

## Input

第一行为两个正整数  $n, k (1 \leq k \leq n \leq 100)$  分别表示有几个地区和背包容量大小  
第二行为  $n$  个正整数  $a_1, a_2, \dots, a_n (1 \leq a_i \leq n)$ ，表示第  $i$  个地区的机器需要使用第  $a_i$  类工具维修  
第三行为  $n$  个整数  $b_1, b_2, \dots, b_n (0 \leq b_i \leq 10^8)$ , 表示第  $i$  种的工具需要  $b_i$  个绿宝石

## Output

输出一个整数表示*ManInM00N*最少需要花多少绿宝石



Sample

standard input	standard output
4 2 1 2 3 1 1 1 1 1	3
7 2 1 2 3 1 1 1 2 1 1 0 1 0 0 0 0	13

Hint

样例 1 中,

- $i = 1$  时, 需要第 1 种工具来维修, 此时背包为空, 没有第 1 种工具, 故花费  $b_1$  个绿宝石购买第 1 种工具
- $i = 2$  时, 需要第 2 种工具来维修, 此时背包为  $\{ 1 \}$ , 没有第 2 种工具, 故花费  $b_2$  个绿宝石购买第 2 种工具
- $i = 3$  时, 需要第 3 种工具来维修, 此时背包为  $\{ 1, 2 \}$ , 没有第 3 种工具且背包满了, 我们选择丢弃第 2 种工具一把, 花费  $b_3$  购买第 3 种工具
- $i = 4$  时, 需要第 1 种工具来维修, 此时背包为  $\{ 1, 3 \}$ , 可以不用购买工具

此时最少的花费为 3 ,如果在  $i = 3$  时, 选择丢弃第 1 种工具, 则需要在  $i = 4$  时多花费  $b_1$  个绿宝石

# 11: 方块世界の魔法小径（easy version）

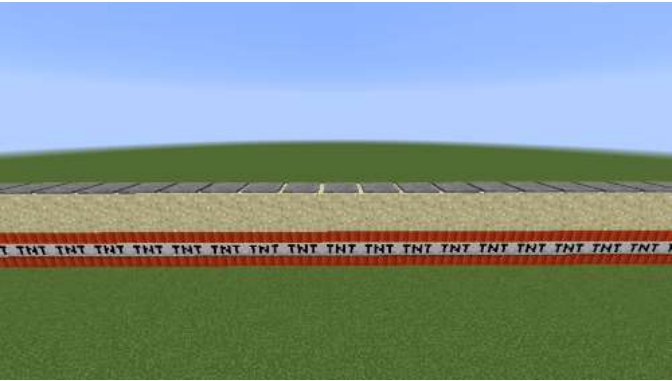
Time Limit: 1000MS

Memory Limit: 32268KB

Description

注意：本题easy和hard版本中只有n和时间限制不同

SevenGod, ManInM00N, Taibo三位方块人在冒险途中，找到了一条神秘的魔法小道，这条道路上有n个压力板，编号为1, 2, ..., n，每个压力板下放着一个沙子和TNT，如图所示。



经过反复的游玩，三人还发现了这条道路的几个限制

- 初始你在第1个压力板上
- 当踩到第*i*个压力板上时会获得 $A_i$ 层跳跃提升的buff，同时玩家拥有无限的力量药水，每次跳跃前可以选择一瓶力量 $x$ （ $x$ 为正整数）的药水，使自己此次能够跳跃 $A_i \times x$ 的距离，假设当前位置为*i*，跳跃后会到达 $i + A_i \times x$ 的位置，这是你唯一可以进行的操作。
- 所有你踩过的压力板 $p_1, p_2, \dots, p_K$ 都会塌落， $\{p_1, p_2, \dots, p_k\}$ 是你踩的压力板的编号集合。
- 当你喝下一瓶力量药水导致你跳出n个压力板范围时，游戏结束。即 $i + A_i \times x > n$ 时游戏结束。

聪明的Taibo很快就玩腻了，但是身为数学天才的他想到了一个有趣的问题并以此来考考他的伙伴。

Taibo：求游戏结束时所有可能掉落的方块的集合数，由于数量过多，请模998244353后输出。

ManInM00N :

22计算机3陈忠鹏 LV39 群主

SevenGod:

既然如此，就交给屏幕前的各位吧

Input

第一行输入一个整数 $N$

$1 \leq N \leq 1 \times 10^3$

第二行输入 $N$ 个整数 $A_1, A_2 \dots A_N$

$1 \leq A_i \leq 2 \times 10^5$

Output

输出一个整数 $x$ ，表示掉落的方块的集合数为 $x$

Sample

standard input	standard output
5 1 2 3 1 1	8

Hint

八种掉落方块的可能方式为

$\{1\}, \{1, 2\}, \{1, 2, 4\}, \{1, 2, 4, 5\}, \{1, 3\}, \{1, 4\}, \{1, 4, 5\}, \{1, 5\}$

举个例子，对于 $\{1, 4, 5\}$ ，步骤如下

- 初始位于位置1，选择力量3，跳到 $1 + 3 \times 1 = 4$ 的位置
- 位于位置4，选择力量1，跳到 $4 + 1 \times 1 = 5$ 的位置
- 位于位置5，无论如何都会跳出界外，游戏结束，踩过的压力表编号为 $\{1, 4, 5\}$ ，为一个可能的解

# 12: 方块世界の魔法小径（hard version）

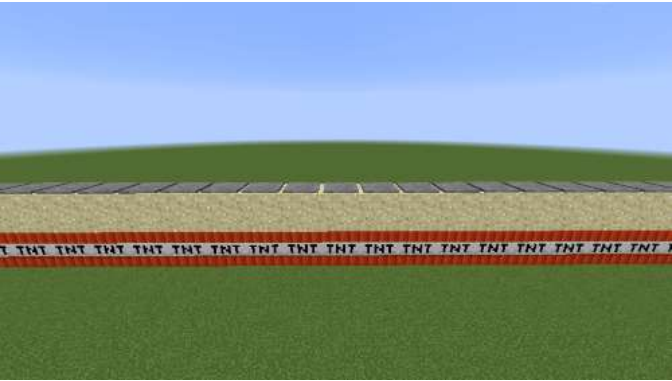
Time Limit: 5000MS

Memory Limit: 32268KB

Description

注意：本题easy和hard版本中只有n和时间限制不同

SevenGod, ManInM00N, Taibo三位方块人在冒险途中，找到了一条神秘的魔法小道，这条道路上有n个压力板，编号为1, 2, ..., n，每个压力板下放着一个沙子和TNT，如图所示。



经过反复的游玩，三人还发现了这条道路的几个限制

- 初始你在第1个压力板上
- 当踩到第*i*个压力板上时会获得 $A_i$ 层跳跃提升的buff，同时玩家拥有无限的力量药水，每次跳跃前可以选择一瓶力量 $x$ （ $x$ 为正整数）的药水，使自己此次能够跳跃 $A_i \times x$ 的距离，假设当前位置为*i*，跳跃后会到达 $i + A_i \times x$ 的位置，这是你唯一可以进行的操作。
- 所有你踩过的压力板 $p_1, p_2, \dots, p_K$ 都会塌落， $\{p_1, p_2, \dots, p_k\}$ 是你踩的压力板的编号集合。
- 当你喝下一瓶力量药水导致你跳出n个压力板范围时，游戏结束。即 $i + A_i \times x > n$ 时游戏结束。

聪明的Taibo很快就玩腻了，但是身为数学天才的他想到了一个有趣的问题并以此来考考他的伙伴。

Taibo：求游戏结束时所有可能掉落的方块的集合数，由于数量过多，请模998244353后输出。

ManInM00N :

22计算机3陈忠鹏 LV39 群主

SevenGod:

既然如此，就交给屏幕前的各位吧

Input

第一行输入一个整数 $N$

$1 \leq N \leq 2 \times 10^5$

第二行输入 $N$ 个整数 $A_1, A_2 \dots A_N$

$1 \leq A_i \leq 2 \times 10^5$

Output

输出一个整数 $x$ ，表示掉落的方块的集合数为 $x$

Sample

standard input	standard output
5 1 2 3 1 1	8

Hint

八种掉落方块的可能方式为

$\{1\}, \{1, 2\}, \{1, 2, 4\}, \{1, 2, 4, 5\}, \{1, 3\}, \{1, 4\}, \{1, 4, 5\}, \{1, 5\}$

举个例子，对于 $\{1, 4, 5\}$ ，步骤如下

- 初始位于位置1，选择力量3，跳到 $1 + 3 \times 1 = 4$ 的位置
- 位于位置4，选择力量1，跳到 $4 + 1 \times 1 = 5$ 的位置
- 位于位置5，无论如何都会跳出界外，游戏结束，踩过的压力表编号为 $\{1, 4, 5\}$ ，为一个可能的解

# 13: 第二名>\_<

Time Limit: 1000MS

Memory Limit: 32268KB

## Description

第二名>\_<

某天, *Lococ* 和 *ManInM00N* 双排游玩斗魂竞技场, 其中 *ManInM00N* 获得了飞升护符, 可以通过随机 *roll* 点获取强大的属性。在决胜局的时候, *ManInM00N* 玩的ad刺客觉得护符提供的攻击力和穿甲太少了, 决定用最后一次机会 *roll* 出一个“更好”的属性拿下第一名, 不出意外的话, 是出意外了。



于是, 他喜提第二名

赛后复盘

*Lococ*: “这怎么会输呢? ”,

*ManInM00N*: “他们太胡了吧 (汗”

自这把对局以后, *ManInM00N* 后面的每一把都是第二名。

## Input

本题没有输入

## Output

输出一个 **整数** 表示 *ManInM00N* roll到的多少 **法术强度**

## Sample

standard input	standard output
这题没有输入	输出图中的法术强度的值

## Hint

诶 这是什么？ hello world!

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    cout<<"Hello world";
}
```

你已经学会输出Hello world了，那如何输出任意你想输出的东西呢？

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    cout<<"Legendary Grandmaster";
}
```

哇塞，这样就行了！