

## Problem A. 口算训练

Input file: `stdin`  
Output file: `stdout`  
Time limit: 5 seconds  
Memory limit: 512 megabytes

小 Q 非常喜欢数学，但是他的口算能力非常弱。因此他找到了小 T，给了小 T 一个长度为  $n$  的正整数序列  $a_1, a_2, \dots, a_n$ ，要求小 T 抛出  $m$  个问题以训练他的口算能力。

每个问题给出三个正整数  $l, r, d$ ，小 Q 需要通过口算快速判断  $a_l \times a_{l+1} \times \dots \times a_{r-1} \times a_r$  是不是  $d$  的倍数。

小 Q 迅速地回答了出来，但是小 T 并不知道正确答案是什么，请写一个程序帮助小 T 计算这些问题的正确答案。

### Input

第一行包含一个正整数  $T (1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含两个正整数  $n, m (1 \leq n, m \leq 100000)$ ，分别表示序列长度以及问题个数。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 100000)$ ，表示序列中的每个数。

接下来  $m$  行，每行三个正整数  $l, r, d (1 \leq l \leq r \leq n, 1 \leq d \leq 100000)$ ，表示每个问题。

### Output

对于每个问题输出一行，若是倍数，输出“**Yes**”，否则输出“**No**”。

### Examples

stdin	stdout
1	Yes
5 4	No
6 4 7 2 5	No
1 2 24	Yes
1 3 18	
2 5 17	
3 5 35	

## Problem B. 寻宝游戏

Input file: `stdin`  
 Output file: `stdout`  
 Time limit: 2 seconds  
 Memory limit: 512 megabytes

小 Q 最近迷上了一款寻宝游戏，这款游戏中每局都会生成一个  $n \times m$  的网格地图，从上往下依次编号为第 1 行到第  $n$  行，从左往右依次编号为第 1 列到第  $m$  列。每个格子上都有不同数量的金币，第  $i$  行第  $j$  列的格子上的金币数量为  $a_{i,j}$ 。

小 Q 一开始位于  $(1,1)$ ，每次他可以往右或者往下走，每当他经过某个格子时，他就可以拿走这个格子上的所有金币。小 Q 不能走出这个地图，当小 Q 不能再行动时，游戏结束。显然当且仅当小 Q 位于  $(n,m)$  时，游戏才会结束。

一轮游戏的得分为这一轮中收集到的金币总量，而在游戏开始前，因为小 Q 是超级 VIP 用户，所以他有  $k$  次机会交换某两个格子中的金币数。这  $k$  次机会不一定要用完，请写一个程序帮助小 Q 在一轮内拿到尽可能多的金币。

### Input

第一行包含一个正整数  $T(1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含三个整数  $n, m, k(2 \leq n, m \leq 50, 0 \leq k \leq 20)$ ，分别表示地图的长宽以及交换的次数。

接下来  $n$  行，每行  $m$  个整数  $a_{i,j}(0 \leq a_{i,j} \leq 10^6)$ ，依次表示每个格子中金币的数量。

### Output

对于每组数据，输出一行一个整数，即能收集到的金币数量的最大可能值。

### Examples

stdin	stdout
2	34
3 4 0	81
1 2 3 4	
9 8 7 6	
5 4 7 2	
5 5 1	
9 9 9 0 0	
0 0 9 0 0	
0 0 0 0 0	
0 0 9 0 0	
9 0 9 9 9	

## Problem C. 二叉排序树

Input file: `stdin`  
Output file: `stdout`  
Time limit: 20 seconds  
Memory limit: 512 megabytes

二叉排序树或者是一棵空树，或者是具有下列性质的二叉树：

- 若左子树不空，则左子树上所有节点的值均小于或等于它的根节点的值。
- 若右子树不空，则右子树上所有节点的值均大于或等于它的根节点的值。
- 左、右子树也分别为二叉排序树。

我们考虑按 1 到  $n$  的顺序依次往一棵空二叉排序树中插入  $n$  个互不相同的数  $a_1, a_2, \dots, a_n$ 。插入数字  $x$  的算法如下：从根开始，如果节点为空，则将其值设置为  $x$  并退出；否则若  $x$  小于根的值，则往左儿子递归插入，反之往右儿子递归插入。

令小 Q 感到惊讶的是，对于给定的数列  $a_1, a_2, \dots, a_n$ ，将该数列顺序打乱后再从左往右依次插入  $a_1, a_2, \dots, a_n$ ，最终可能会得到一模一样的二叉排序树。比如  $(2, 3, 1)$  和  $(2, 1, 3)$  就会得到相同的结果。

请写一个程序，帮助小 Q 找到对于给定的  $k$ ，是否存在一个数列  $a_1, a_2, \dots, a_n$ ，使得恰好有  $k$  种插入顺序得到的结果与按  $1, 2, \dots, n$  的顺序插入得到的结果完全相同。

### Input

第一行包含一个正整数  $k(1 \leq k \leq 10^9)$ 。

### Output

第一行输出一个正整数  $n$ ，表示序列长度，你需要保证  $1 \leq n \leq 4000$ 。

第二行输出  $n$  个正整数  $a_1, a_2, \dots, a_n$ ，你需要保证  $1 \leq a_i \leq n$  且  $a$  中所有数互不相同。

若找不到满足输出要求的数列，请输出一行“-1”；若有多组解，输出任意一组。

### Examples

stdin	stdout
8	5 2 1 4 3 5

### Notes

8 种插入顺序依次为  $(2, 1, 4, 3, 5)$ ,  $(2, 1, 4, 5, 3)$ ,  $(2, 4, 1, 3, 5)$ ,  $(2, 4, 1, 5, 3)$ ,  $(2, 4, 3, 1, 5)$ ,  $(2, 4, 3, 5, 1)$ ,  $(2, 4, 5, 1, 3)$ ,  $(2, 4, 5, 3, 1)$ 。

## Problem D. 对称数

Input file: `stdin`  
 Output file: `stdout`  
 Time limit: 12 seconds  
 Memory limit: 512 megabytes

小 Q 认为，偶数具有对称美，而奇数则没有。

给定一棵  $n$  个点的树，任意两点之间有且仅有一条直接或间接路径。这些点编号依次为 1 到  $n$ ，其中编号为  $i$  的点上有一个正整数  $a_i$ 。

定义集合  $S(u, v)$  为  $u$  点到  $v$  点的唯一最短路径上经过的所有点  $x$  (包括  $u$  和  $v$ ) 对应的正整数  $a_x$  的集合。小 Q 将在  $m$  个  $S(u, v)$  中寻找最小的对称数。因为偶数具有对称美，所以对称数是指那些出现了偶数次 (包括 0 次) 的正整数。

请写一个程序，帮助小 Q 找到最小的对称数。

### Input

第一行包含一个正整数  $T (1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含两个正整数  $n, m (1 \leq n, m \leq 200000)$ ，分别表示点数和询问数。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 200000)$ ，依次表示每个点上的数字。

接下来  $n - 1$  行，每行两个正整数  $u_i, v_i (1 \leq u_i, v_i \leq n, u_i \neq v_i)$ ，表示一条连接  $u_i$  和  $v_i$  的双向树边。

接下来  $m$  行，每行两个正整数  $u_i, v_i (1 \leq u_i, v_i \leq n)$ ，依次表示每个询问。

### Output

对于每个询问输出一行一个正整数，即最小的对称数。

### Examples

stdin	stdout
1	2
5 3	1
1 2 2 1 3	1
1 2	
1 3	
2 4	
2 5	
2 3	
1 4	
2 5	

## Problem E. quailty 算法

Input file:            **stdin**  
 Output file:           **stdout**  
 Time limit:            **2 seconds**  
 Memory limit:         **512 megabytes**

二分图又称作二部图，是图论中的一种特殊模型。设  $G = (V, E)$  是一个无向图，如果点集合  $V$  可分割为两个互不相交的子集  $A, B$ ，并且图中的每条边  $(i, j)$  所关联的两个顶点  $i$  和  $j$  分别属于这两个不同的顶点集 ( $i \in A, j \in B$ )，则称图  $G$  为一个二分图。

一个“匹配”是指一组  $E$  的子边集  $M$ ，满足  $M$  中任意两条边都没有公共顶点。“最大匹配”是指  $M$  中边数最多的一个匹配。“最小费用最大匹配”是指在边数最多的基础上，所选择的边的权值总和最小的匹配。

小 T 最近刚刚学会了二分图最大权匹配的 KM(Kuhn-Munkres) 算法，知道 KM 算法稍加修改就能计算二分图最小费用最大匹配。小 T 兴奋地把这个算法讲给小 Q 听，而小 Q 却吹牛说他自己发明了一个 quailty 算法，可以在低于线性的时空复杂度里求解二分图最小费用最大匹配。

小 T 不相信，于是给定  $n$  和一个长度为  $n$  的正整数数组  $a_1, a_2, \dots, a_n$ ，写了一段代码根据  $a[]$  生成了一个很大的带权二分图，代码如下：

```
int ca = n;

long long cb = 0;

for(int i = 1; i <= n; i++){
    for(int j = 1; j < i; j++){
        cb++;

        long long k = cb;

        addedge(i, k, a[i] xor a[j]);

        addedge(j, k, a[i] xor a[j]);
    }
}
```

相关说明：

- $ca$  表示  $A$  中的点数， $cb$  表示  $B$  中的点数，点都是从 1 开始连续编号的。
- “`addedge(x,y,z)`”表示添加一条边权为  $z$  的无向边，两端点分别为  $A$  中的  $x$  点和  $B$  中的  $y$  点。
- “`xor`”表示按位异或的运算符  $\oplus$ 。按位异或的运算符是双目运算符，具有交换律，即  $i \oplus j = j \oplus i$ 。按位异或可以理解成被运算的数字的二进制位对应位如果相同，则结果的该位置为 0，否则为 1，例如： $1(01) \oplus 2(10) = 3(11)$ 。按位异或还可以理解成参与运算的数字的每个二进制位都进行了不进位的加法，例如： $3(11) \oplus 3(11) = 0(00)$ 。

面对如此庞大的图，小 Q 自然不能算出来，于是他向你紧急求助。请写一个程序，求出最大匹配所需的最小费用。

## Input

第一行包含一个正整数  $T(1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含一个正整数  $n(1 \leq n \leq 300000)$ 。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^9)$ 。

## Output

对于每组数据，输出一行一个整数，即最大匹配所需的最小费用。

## Examples

stdin	stdout
2	30
3	20
8 2 5	
5	
9 3 5 7 1	

## Problem F. 激光实验

Input file: `stdin`  
Output file: `stdout`  
Time limit: 10 seconds  
Memory limit: 512 megabytes

小 T 在他的激光发射器中加入了智能芯片，并强化了激光的威力，正在一个实验场上进行激光发射实验。

实验场是一个  $n \times m$  的网格地图，从下往上依次编号为第 1 行到第  $n$  行，从左往右依次编号为第 1 列到第  $m$  列。除了左下角  $(1,1)$  外，每个格子内部恰好有一个靶子。

小 T 将它的激光发射器固定在  $(1,1)$ ，面向东方 (即  $(1,2)$ )。智能芯片将会控制发射器依次击毁全部靶子，具体如下：

- 如果激光发射器朝向的方向存在至少一个未被击毁的靶子，则发射器会发射一束激光摧毁该方向最近的未被摧毁的靶子。注意靶子不可穿透，所以需要分多次来摧毁该方向的所有靶子。
- 如果该方向所有靶子都已摧毁，则逆时针旋转朝向，直到面朝的方向至少存在一个未被击毁的靶子。

显然，激光发射器一共会发射  $n \times m - 1$  束激光，小 T 想记录  $q$  个实验数据，那就是第  $k_i$  个被摧毁的靶子的坐标应该是多少。

请写一个程序帮助小 T 计算这  $q$  个实验数据的预计数值。

### Input

第一行包含三个正整数  $n, m, q$  ( $1 \leq n, m \leq 10^9, \max(n, m) > 1, 1 \leq q \leq 1000$ )，表示实验场的尺寸以及询问次数。

接下来  $q$  行，每行一个正整数  $k_i$  ( $1 \leq k_i < n \times m$ )，依次表示每个询问。

### Output

对于每个询问输出一行两个正整数  $r, c$ ，表示靶子坐标为  $(r, c)$ 。

### Examples

stdin	stdout
3 5 3	1 2
1	3 1
14	3 5
8	

## Problem G. 摘抄大赛

Input file: `stdin`  
 Output file: `stdout`  
 Time limit: 6 seconds  
 Memory limit: 512 megabytes

比特镇一年一度的摘抄大赛开始啦！今年一共有  $n$  位选手参赛，编号依次为 1 到  $n$ 。比赛规则是：主办方给出一个小写字母组成的字符串  $S[1..len]$ ，每位选手在规定时间内从中选取一个连续子串  $S[l, r]$  摘抄。最终将所有选手按照摘抄的子串的字典序从小到大排列，对于字典序相同的选手，则按编号从小到大排列。

比较两个字符串的字典序，是从第 1 位开始，若不相同则判断这一位较小的那个串字典序更小（认为空串小于一切），若相同则接着比较下一位，直至得出结果为止。

因为参赛人数实在是太多了，主办方忙得不可开交。请写一个程序帮助主办方将这些选手按照规则排列。

### Input

第一行包含一个正整数  $T(1 \leq T \leq 20)$ ，表示测试数据的组数。

每组数据第一行包含两个正整数  $n, m(1 \leq n \leq 300000, 1 \leq m \leq 2000)$ ，其中  $n$  表示参赛选手的数量， $m$  表示压缩后的字符串信息段数。

接下来  $m$  行，每行一个小写字母  $a_i$  和一个正整数  $b_i(1 \leq b_i \leq 10^{14})$ ，表示字符  $a_i$  连续重复  $b_i$  次。最终  $S$  由这  $m$  段字符串依次拼接得到，如  $[a, 3], [b, 4], [c, 1], [c, 5] = aaabbbbcccccc$ ，因此  $len = \sum b_i$ 。

接下来  $n$  行，每行两个正整数  $l_i, r_i(1 \leq l_i \leq r_i \leq len)$ ，分别表示每个选手选择摘抄的子串。

### Output

对于每组数据，输出一行  $n$  个正整数，即按规则排列后的选手名单，第  $i$  个数为排在第  $i$  位的选手编号。注意最后一个数之后不要有空格。

### Examples

stdin	stdout
1	2 3 1
3 4	
a 3	
b 4	
c 1	
c 5	
1 3	
1 2	
1 2	



## Problem H. 回文树

Input file: `stdin`  
 Output file: `stdout`  
 Time limit: 3 seconds  
 Memory limit: 512 megabytes

给定一棵  $n$  个点的树，任意两点之间有且仅有一条直接或间接路径。这些点编号依次为 1 到  $n$ ，其中编号为  $i$  的点上有一个正整数  $a_i$ 。你可以认为每个数  $a_i$  都是在  $[1, n]$  里等概率随机挑选的。

令  $S(u, v)$  表示在树上  $u$  到  $v$  的唯一最短路径上，按照离  $u$  从近到远依次考虑每个点，将它们上面的数字  $a$  按顺序写下连成的字符串。若一个字符串正着读和倒着读相等，则称它为回文串，比如“1 2 1”和“1 3 3 1”是回文串，但是“1 2 1”不是回文串。

请写一个程序，统计有多少个正整数对  $(u, v)$  满足  $1 \leq u \leq v \leq n$ ，且  $S(u, v)$  是回文串。

### Input

第一行包含一个正整数  $T(1 \leq T \leq 20)$ ，表示测试数据的组数。

每组数据第一行包含一个正整数  $n(1 \leq n \leq 100000)$ ，表示点数。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq n)$ ，依次表示每个点上的数字。你可以认为每个数  $a_i$  都是在  $[1, n]$  里等概率随机挑选的。

接下来  $n - 1$  行，每行两个正整数  $u_i, v_i(1 \leq u_i, v_i \leq n, u_i \neq v_i)$ ，表示一条连接  $u_i$  和  $v_i$  的双向树边。

### Output

对于每组数据，输出一行一个整数，即满足条件的点对  $(u, v)$  的数量。

### Examples

stdin	stdout
2	4
3	3
1 2 1	
1 2	
2 3	
2	
2 2	
1 2	

## Problem I. 代码派对

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 512 megabytes

比特镇的算法竞赛选手们应邀来到了小 Q 家中参加盛大的“代码派对”。小 Q 计划在家中举行一场 ACM 赛制的友谊赛，3 个人为一队进行比赛。

为了让比赛更加有趣，也为了让大家能多多了解他人，小 Q 在花园的地上画出了一个  $1000 \times 1000$  的网格图，从上往下依次编号为第 1 行到第 1000 行，从左往右依次编号为第 1 列到第 1000 列。小 Q 让每一名选手选择一个网格图内平行坐标轴的子矩形  $(x_1, y_1) - (x_2, y_2)$ ，表示以  $(x_1, y_1)$  和  $(x_2, y_2)$  为对顶点的矩形，包含所有满足  $x \in [x_1, x_2]$  且  $y \in [y_1, y_2]$  的格子。

小 Q 规定， $i, j, k (1 \leq i < j < k \leq n)$  三人组成一支三人队伍能参赛，当且仅当这三个人选择的子矩形包含至少一个公共格子。请写一个程序，帮助小 Q 统计有多少种三人组合组成的队伍可以参赛。

### Input

第一行包含一个正整数  $T (1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含一个正整数  $n (3 \leq n \leq 100000)$ ，表示选手人数。

接下来  $n$  行，每行四个正整数  $x_1, y_1, x_2, y_2 (1 \leq x_1 \leq x_2 \leq 1000, 1 \leq y_1 \leq y_2 \leq 1000)$ ，分别表示每个选手选择的矩形的顶点坐标。

### Output

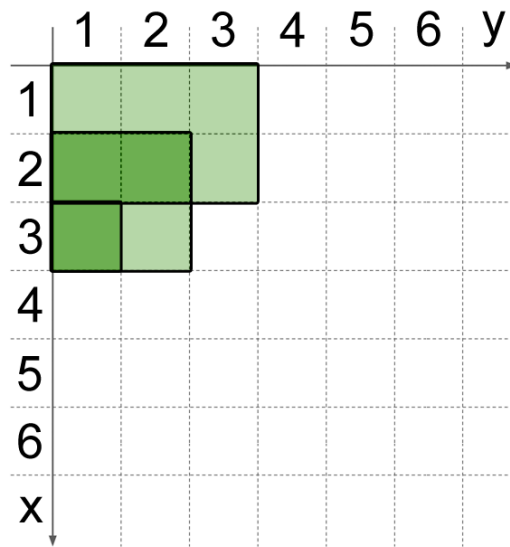
对于每组数据，输出一行一个整数，即满足条件的三人组合数量。

### Examples

stdin	stdout
2	0
3	4
3 1 3 1	
1 1 2 3	
2 1 3 2	
5	
1 1 4 5	
2 1 3 2	
2 2 3 3	
4 5 4 5	
1 2 2 4	

## Notes

第一组数据如下图：



第二组数据如下图：

