

Problem A. 沙币

Input file: standard input
Output file: standard output
Time limit: 15 seconds
Memory limit: 16 megabytes

虚拟货币现在是一项很赚钱的买卖, 小gie也希望能分一杯羹。他打算尝试挖一下一种PoW共识机制的虚拟货币——沙币。

但是他并不想买设备或者自己出电费。他想把要求解的难题发给别人, 薅别人的电费和算力来帮他解决。现在他找上了你。

共有 q 个问题, 每个问题包含一个数字 y 和一个难度因子 w 。你需要找到一个 x , 使得 $(f(x) \text{ xor } y) \text{ shr } (32 - w) = 0$ (xor表示按位异或, shr表示逻辑右移), 即 $f(x)$ 和 y 的二进制表示中高 w 位相等。 $f(x)$ 等价于如下的c++代码:

```
uint32_t f(uint32_t x) {  
    x ^= x << 13;  
    x ^= x >> 17;  
    x ^= x << 5;  
    return x;  
}
```

由于小gie有强迫症, 如果有多个 x 存在, 他希望你输出最小的那个。

为了防止你找 $f(x)$ 的性质, 小gie提醒你, $f(x)$ 没有什么有用的性质。因为如果一个人能够找到一些 $f(x)$ 的性质, 就能更快地求解出所需的 x , 使得工作量证明机制失效。

请注意题目的内存限制!

Input

第一行包含一个数字 $q(1 \leq q \leq 500)$, 表示询问组数。

接下来 q 行, 每行包含两个数字 $y, w(0 \leq y < 2^{32}, 1 \leq w \leq 25)$, 表示要求解的询问。

Output

q 行, 每行一个数字 x , 表示待求解的询问的答案。显然, 答案一定存在。

Example

standard input	standard output
5	0
599274826 1	4096
1088402927 2	4096
1101806161 3	4096
1192646568 4	1536
450755019 5	

Problem B. 组队

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

有 $3 \times N$ 位参赛者参加了2020北京邮电大学校赛, 其中第 i 位参赛者的实力为 a_i 。 $3 \times N$ 位参赛者组成 N 支队伍, 每支队伍有3名参赛者, 每支队伍的實力定义为队伍中實力排名第二的成員的實力。为体现学校实力, 教练要给这些参赛者搭配组队, 使得 N 支队伍的实力的和尽量大, 请求出队伍实力和的最大值。

Input

第一行一个整数 $N(1 \leq N \leq 10^5)$ 。

接下来一行 $3 \times N$ 个整数 $a_i(1 \leq a_i \leq 2^{31} - 1)$ 。

Output

输出一个整数, N 支队伍的實力和的最大值。

Example

standard input	standard output
2 5 2 8 5 1 5	10

Problem C. 染色

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

有一张 N 个点 M 条边的无向图, 所有边的长度都是1。每个点会有一个颜色, 初始时每个点的颜色编号是0。

会有 Q 个操作来对该图进行染色, 每个操作会有三个参数 v, d, c 表示将距离 v 点不超过 d 的所有点颜色变成 c 。此处, 两点之间的距离定义为它们之间的最短路上的边数。

现求 Q 个操作后每个点的颜色。

Input

第一行两个整数 $N, M(1 \leq N, M \leq 2 \times 10^5)$ 。

接下来 M 行, 每行两个整数 $a_i, b_i(1 \leq a_i \leq N, 1 \leq b_i \leq N)$ 。

接下来一个整数 $Q(1 \leq Q \leq 2 \times 10^5)$ 。

最后 Q 行, 每行三个整数 $v_i, d_i, c_i(1 \leq v_i \leq N, 0 \leq d_i \leq 20, 1 \leq c_i \leq 2 \times 10^5)$ 。

Output

输出 N 行, 每行一个整数, 第 i 行输出的是 N 次操作后第 i 个点的颜色。

Example

standard input	standard output
7 7	2
1 2	2
1 3	2
1 4	2
4 5	2
5 6	1
5 7	0
2 3	
2	
6 1 1	
1 2 2	

Problem D. DDL

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

小A是个DDL战士, 对于每一项作业, 小A总是希望在最接近DDL (作业截止时间) 的时候将其完成。
具体地说, 每一项作业有一个布置日期 S_i 和截止日期 T_i , 小A要求自己完成作业的日期 F_i 满足 $S_i \leq F_i \leq T_i$, 而且小A每天只能完成一项作业。同时, 在完成作业 i 时, 小A会由于完成日期 F_i 小于DDL而产生一个悲痛值 P_i , 具体的计算方式为 $P_i = T_i - F_i$ 。

你的任务是帮助小A确定每一项作业的完成日期, 以保证在每项在规定时间内完成的前提下, 悲痛值的和尽量小。

Input

第一行, 一个正整数 N ($1 \leq N \leq 10^5$), 表示作业的数量。

之后的 N 行, 每行输入两个正整数, S_i 和 T_i ($1 \leq S_i \leq T_i \leq 10^9$), 每个作业的布置日期和截止日期。

Output

若没有一种方案可以使小A在规定时间内完成所有作业则输出 -1 。否则输出悲痛值的和。

Example

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

Problem E. 栽树

Input file: standard input
Output file: standard output
Time limit: 10 seconds
Memory limit: 512 megabytes

尼采家财万贯，家里有万亩良田。他不用这些田来种庄稼，而用来栽树——嘿，就是玩儿。他的树林可以看做一个 $n \times m$ 的矩阵。每棵树都有一定的高度。现在尼采想要对树林进行魔改。他每次会选择两个形状和大小相同的矩形，然后将这两个矩形中对应位置的树木相互交换。他想知道经过 q 次交换，每个位置的树木的高度是多少。

尼采保证每次选择的两个矩形相互不重叠且不相邻，即没有一棵树同时属于两个矩形，也没有两棵相邻的树分别属于两个矩形。

Input

第一行三个整数 n, m, q ($1 \leq n, m \leq 3 \times 10^3, 0 \leq q \leq 10^4$)，表示树林的长和宽以及魔改的次数。

接下来 n 行，每行 m 个整数，表示开始时每棵树的高度 $L_{i,j}$ ($0 \leq L_{i,j} \leq 10^7$)。

接下来 q 行，每行六个整数 $a_i, b_i, c_i, d_i, h_i, w_i$ ，其中 $(a_i, b_i), (c_i, d_i)$ 分别为两个矩形的左上角， h_i, w_i 分别为矩形的长和宽，即两个矩形的右下角分别为 $(a_i + h_i - 1, b_i + w_i - 1)$ 和 $(c_i + h_i - 1, d_i + w_i - 1)$ 。

保证输入数据小于20MB。

Output

共 n 行，每行 m 个整数，表示经过修理后的每棵树的高度。

Example

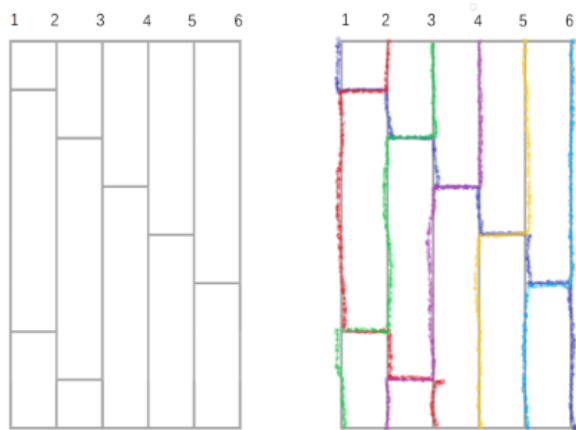
standard input	standard output
4 4 2	4 4 3 3
1 1 2 2	4 4 3 3
1 1 2 2	2 2 1 1
3 3 4 4	2 2 1 1
3 3 4 4	
1 1 3 3 2 2	
3 1 1 3 2 2	

Problem F. 排列

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

尼采想出了一种生成1到 n 的排列的一种方法。

他首先画出了一个矩形框，并在里面画出一些竖线。如图所示，图里共有六条竖线。他从左到右依次编号1到 n 。接下来，他在两个相邻的竖线之间划横线。在同一高度只会画一条横线。每个数字会沿着它所在的直线下降，如果碰到了横线，就走到横线的另外一端，然后继续下降。可以证明，每个数字都会最后停在不同的竖线上，因此这种构造方法就生成了一个排列。



现在尼采想给你一张图，让你告诉他最终生成的排列。同时他还想知道，如果让你设计一张图生成同样的排列，至少需要画多少道横线。

Input

第一行两个整数 n, m ($1 \leq n \leq 3 \times 10^5, 1 \leq m \leq 10^6$), 表示序列长度和横线的数量。

第二行 m 个数字，表示从上到下每个横线的位置。输入 a 表示有一条横线连接第 a 和第 $a + 1$ 条竖线, $1 \leq a < n$ 。

Output

第一行 n 个整数，表示最终的排列。

第二行一个整数，表示得到同样的排列最少需要多少条横线。

Example

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

Problem G. 修复

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

小A的硬盘坏了, 于是他问万能的小B怎么把硬盘修好, 小B仔细的研究了这块硬盘, 然后建议小A换一块新的。可是小A舍不得这块硬盘, 因为这里面有很多**好康的**。于是小A决定自己把这块硬盘修好。

具体地说, 硬盘被简化为 N 个存储单元($N \leq 500$), 每个存储单元有一个编号 $i(1 \leq i \leq N)$ 。所谓的硬盘坏了, 就是指硬盘的 N 个存储单元中有一些发生损坏, 而其他的则完好无损。在之前的检查中, 小B告诉小A有 $M(M \leq N)$ 个存储单元发生损坏, 但是小A不知道这 M 个存储单元的具体编号。于是小A下面的任务就是找到哪些编号的单元是损坏的。

小A将尝试使用自己的魔法来检测硬盘, 小A每次选择一个正整数 X 然后发动魔法, 魔法的效果是询问编号小于等于 X 的单元中有没有**未被找到的**损坏单元, 发动魔法后他将获得回答: 是或否。当小A能确定某个编号的单元是损坏的时, 他会顺手把这个单元修好。

你的任务是, 在小A每次询问时, 帮他选择每次询问的 X , 使得小A询问的次数在最坏的情况下最少。

Input

两个正整数 $N, M(M \leq N \leq 500)$, 用一个空格隔开

Output

一个正整数, 表示在最坏情况下小A的最少询问次数

Examples

standard input	standard output
5 2	4
5 1	3

Note

需要注意的是, 若所有单元都已损坏, 则显然小A不需要进行任何询问就可以确定哪些单元是损坏的

Problem H. 区间

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

给定两个整数 A 和 B , 判断能否找到一个区间 $[l, r]$ 使得区间中存在 A 个奇数和 B 个偶数。

Input

第一行包含一个整数 $T(1 \leq T \leq 10^6)$, 表示询问组数。

接下来 T 行, 每行包含两个整数 $A, B(0 \leq A, B \leq 10^{18})$ 。

Output

对于每组数据, 如果不存在, 则输出GG, 否则输出MM。

Example

standard input	standard output
2	MM
2 3	GG
5 8	