

Problem A. 挑战 NP-Hard

Input file: `stdin`
Output file: `stdout`
Time limit: 1 seconds
Memory limit: 256 megabytes

天才大学生 quailty 热衷于解决 NP-Hard 问题，你如果 AC 了这道题，就可以成为他真正的粉丝。

图染色问题：给定无向图 G 和一个正整数 k 。对于图中的每个点，选择一个在 $[1, k]$ 之间的整数作为其颜色。你需要保证对于每条边，其两端点的颜色均不相同。

简单 k 路径问题：给定无向图 G 和一个正整数 k 。请找到一条经过了恰好 k 条边的简单路径。即，你需要找到一个长度为 $k + 1$ 的序列 v_1, v_2, \dots, v_{k+1} ，满足 $1 \leq v_i \leq n$ ，且任意两个 v 均不相同，同时 v_i 与 v_{i+1} 之间存在一条边。

现在给定无向图 G 和一个正整数 k ，quailty 知道你没有他的水平，所以你只需解决上面的任意一个问题就可以成为他的粉丝。

Input

第一行包含一个正整数 $T (1 \leq T \leq 1000)$ ，表示测试数据的组数。

对于每组数据，第一行包含三个正整数 $n, m, k (1 \leq n \leq 1000, 1 \leq m \leq 10000, 1 \leq k \leq n)$ ，分别表示图的点数与边数。

接下来 m 行，每行两个正整数 $a, b (1 \leq a, b \leq n)$ ，表示 a 到 b 之间存在一条无向边。

输入数据保证不存在重边与自环，且总边数不超过 100000。

Output

对于每组数据：

若选择了图染色问题，请输出 “color”，然后输出 n 个在 1 到 k 之间的正整数，分别表示每个点的颜色。

若选择了简单路径问题，请输出 “path”，然后输出 $k + 1$ 个在 1 到 n 之间的正整数，分别表示路径上每个点的编号。

若有多组可行解，输出任意一组。

Examples

stdin	stdout
2	path 3 2 1
4 5 2	color 1 2 3
1 2	
2 3	
3 4	
4 1	
1 3	
3 3 3	
1 2	
2 3	
3 1	

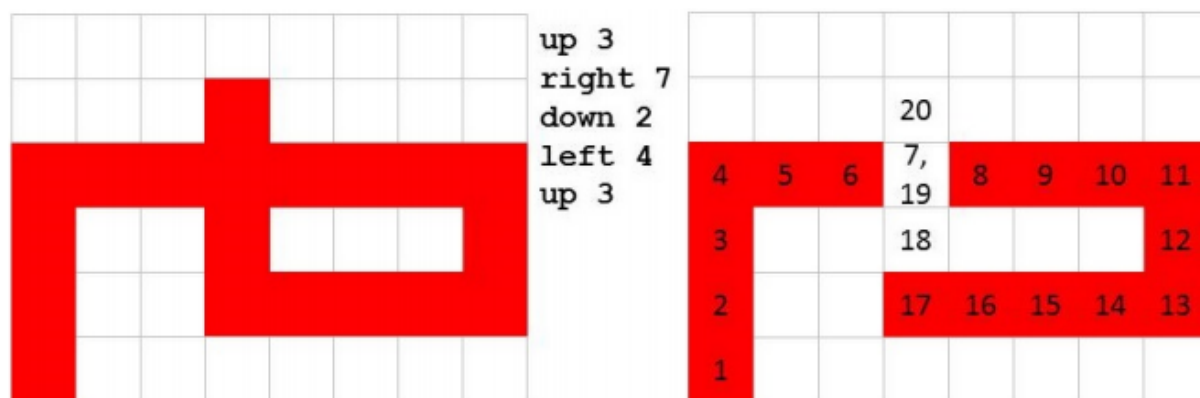
Problem B. 失控的数位板

Input file: stdin
 Output file: stdout
 Time limit: 30 seconds
 Memory limit: 256 megabytes

绘画家小 Q 正在他的数位板上作画。当他在作画的时候，他的设备突然故障了！此时他的笔变成了擦除模式。

在小 Q 的脑海中，有一副漂亮的图案，他希望最后能将它呈现在板子上。他按照自己的创作风格，一步一步地绘画。他每一步只可能是 4 种指令上 (up) 下 (down) 左 (left) 右 (right) 中的一种，同时指定一个距离 d ，每秒往那个方向移动一步。在创作的一开始，他的笔位于数位板的左下角，且任意时刻他的笔都不会离开数位板。

考虑 6×8 的数位板以及如下图所示的指令集，如果笔在第 17 秒末故障了，数位板最终会变成第二幅图的样子。



小 Q 想知道最早和最晚的时刻 t ，满足如果数位板在第 t 秒末故障了，最终显示出来的图案与他预想的一致。注意数位板可以在第 0 秒末故障，此时小 Q 的笔还没有触碰到数位板。

Input

第一行包含三个正整数 h, w, n ($1 \leq h, w, n \leq 10^6, 1 \leq w \times h \leq 10^6$)，分别表示数位板的高度与宽度以及指令的条数。

接下来 h 行，每行 w 个字符，每个字符要么是 ‘#’，要么是 ‘.’。表示小 Q 预想的效果，其中 ‘#’ 表示有颜色，‘.’ 表示空白。

接下来 n 行，每行一个字符串 s 和一个正整数 d ($1 \leq d \leq 10^6$)，分别表示方向和距离，输入数据保证操作不会使笔离开数位板。

Output

输出一行两个整数 x 与 y ，其中 x 表示最小的合法的 t ， y 表示最大的合法的 t ，注意 y 不能超过最后一条指令执行完毕时的时间。若无解则输出 “-1 -1”。

Examples

stdin	stdout
6 8 5 ###.#### #.# #..##### #..... up 3 right 7 down 2 left 4 up 3	17 17

Problem C. 排名的战争

Input file: `stdin`
Output file: `stdout`
Time limit: 8 seconds
Memory limit: 256 megabytes

小 Q 是一名出色的质检员，他负责质检一批手机的质量。手机包含两个性能属性：电池寿命 x_1 与坚硬度 x_2 。

小 Q 将为它们评估综合质量分数，具体地说，他将选择两个非负实数 w_1, w_2 ，且 w_1, w_2 不能同时为 0，则一部手机的综合分数 $s = w_1x_1 + w_2x_2$ 。

在评定出所有手机的分数后，小 Q 会把手机按分数从高到低排序，若有多部手机分数相同，他可以将它们随意排列，因此每部手机的排名都是独一无二的。

聪明的你会发现，对于不同的 w 的选定，手机的最终排名可能会大不一样。因此各个公司都会暗中贿赂小 Q，希望他让自己的排名尽量靠前。

现一共有 n 家公司，每家公司提供了一部手机用于质检。tangjz 知道小 Q 可以通过调参来控制排名，所以他想知道他的公司的手机排名最高是多少，最低是多少。

Input

第一行包含一个正整数 $n(1 \leq n \leq 100000)$ ，即公司的个数。

接下来 n 行，每行两个正整数 $x_1, x_2(1 \leq x_1, x_2 \leq 1000)$ ，分别表示每部手机的两个属性。

tangjz 所在公司提供的手机总是输入里的第一部手机。

Output

输出一行两个整数，即最高排名与最低排名。

Examples

stdin	stdout
5 7 7 11 10 8 5 1 1 12 12	3 4

Problem D. 线段游戏

Input file: **stdin**
Output file: **stdout**
Time limit: 3 seconds
Memory limit: 256 megabytes

quailty 和 tangjz 正在玩一个关于线段的游戏。在平面上有 n 条线段，编号依次为 1 到 n 。其中第 i 条线段的两端点坐标分别为 $(0, i)$ 和 $(1, p_i)$ ，其中 p_1, p_2, \dots, p_n 构成了 1 到 n 的一个排列。

quailty 先手，他可以选择一些互不相交的线段，将它们拿走，当然他也可以一条线段也不选。然后 tangjz 必须拿走所有剩下的线段，若有两条线段相交，那么他就输了，否则他就赢了。注意若 quailty 拿走了全部线段，那么 tangjz 也会胜利。

quailty 深深喜欢着 tangjz，所以他不希望 tangjz 输掉游戏，请计算他有多少种选择线段的方式，使得 tangjz 可以赢得游戏。

Input

第一行包含一个正整数 n ($1 \leq n \leq 100000$)，表示线段的个数。

第二行包含 n 个正整数 p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$)，含义如题面所述。

Output

输出一行一个整数，即 tangjz 胜利的方案数，因为答案很大，请对 998244353 取模输出。

Examples

stdin	stdout
5 1 2 4 5 3	8

Problem E. 卡常生成树

Input file: `stdin`
 Output file: `stdout`
 Time limit: 5 seconds
 Memory limit: 256 megabytes

小 Q 最近沉迷底层优化，他终于将他的 MST 模板优化到了很可怕的地步，所以他出了这道题来试验模板的速度。

给定一个 n 个点的带权无向连通图，点的编号从 1 到 n ，请求出这个图的最小生成树。

为了避免输入数据过于庞大，数据进行了压缩，解压方式可用下面这段代码解释：

```

1 void encode(int seed, int n){
2     int x = seed;
3     for(int i = 2; i <= n; i++){
4         x = x * 907 % 2333333;
5         int T = x;
6         for(int j = max(1, i - 5); j <= i - 1; j++){
7             x = x * 907 % 2333333;
8             int w = T ^ x;
9             add_edge(i, j, w); //在点 i 与点 j 之间添加一条边权为 w 的边
10        }
11    }
12 }
```

Input

第一行包含一个正整数 T ($1 \leq T \leq 50$)，表示测试数据的组数。

每组数据只包含一行两个正整数 $n, seed$ ($1 \leq n \leq 10^7, 1 \leq seed \leq 2333332$)，含义如题面所述。

Output

对于每组数据，输出一行一个整数，即最小生成树的树边的权值之和。

Examples

stdin	stdout
3	750887251
3010 2016	3382896
3 14159	9210525875
31415 926	

Problem F. 棋盘上的守卫

Input file: `stdin`
Output file: `stdout`
Time limit: 3 seconds
Memory limit: 256 megabytes

在一个 $n \times m$ 的棋盘上要放置若干个守卫。对于 n 行来说，每行必须恰好放置一个横向守卫；同理对于 m 列来说，每列必须恰好放置一个纵向守卫。

每个位置放置守卫的代价是不一样的，且每个位置最多只能放置一个守卫，一个守卫不能同时兼顾行列的防御。

请计算控制整个棋盘的最小代价。

Input

第一行包含两个正整数 n, m ($2 \leq n, m \leq 100000, n \times m \leq 100000$)，分别表示棋盘的行数与列数。

接下来 n 行，每行 m 个正整数，其中第 i 行第 j 列的数 $w_{i,j}$ ($1 \leq w_{i,j} \leq 10^9$) 表示在第 i 行第 j 列放置守卫的代价。

Output

输出一行一个整数，即占领棋盘的最小代价。

Examples

stdin	stdout
3 4 1 3 10 8 2 1 9 2 6 7 4 6	19

Notes

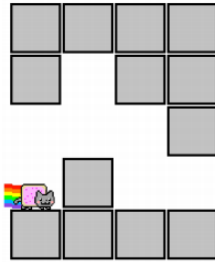
在 (1,1),(2,2),(3,1) 放置横向守卫，在 (2,1),(1,2),(3,3),(2,4) 放置纵向守卫。

Problem G. 太空猫

Input file: `stdin`
 Output file: `stdout`
 Time limit: 1 seconds
 Memory limit: 256 megabytes

太空猫 (SpaceCat) 是一款画面精致、玩法有趣的休闲游戏，你需要控制一只坐在迷你飞碟上的猫咪在太空里不断探索，让大家看看你能飞得多远。

游戏地图可以看成是一个二维的网格图，上下是两段障碍物。在游戏的一开始，太空猫位于地图最左边的下边界之上，且重力方向向下。



在每个时刻，你可以用手指点击屏幕，翻转重力的方向，或者通过遥感控制太空猫往左或往右移动。每次翻转重力方向时，你需要消耗的能量值等于上下底边之间的高度差。在左右移动的时候，太空猫可以下降到对应重力方向更低的位置，但不能往上爬。当然，太空猫也不能穿墙而过。在重力翻转的过程中，直到碰到地面之前，你都不能操控太空猫左右移动。

太空猫的终点位于地图的最右端的下底边之上，请计算为了让太空猫到达终点，需要消耗能量的最小值。

Input

第一行包含一个正整数 n ($1 \leq n \leq 100000$)，即地图的宽度。

第二行包含 n 个正整数 c_1, c_2, \dots, c_n ($2 \leq c_i \leq 10^9$)，分别表示每个横坐标对应的上边界的高度。

第三行包含 n 个正整数 f_1, f_2, \dots, f_n ($1 \leq f_i < c_i$)，分别表示每个横坐标对应的下边界的高度。

Output

输出一行一个整数，即最少的能量，若无法到达终点，请输出“-1”。

Examples

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

Problem H. 长方体

Input file: `stdin`
Output file: `stdout`
Time limit: 10 seconds
Memory limit: 256 megabytes

给定一个 $a \times b \times c$ 的长方体，定义其表面上两个点的距离为沿着长方体的表面走的最短路径的长度，请找到距离最远的点对。

Input

第一行包含三个正整数 $a, b, c (1 \leq a, b, c \leq 1000)$ ，即长方体的长、宽、高。

Output

输出一行一个实数，即最远点对的距离，与标准答案的绝对或相对误差不超过 10^{-8} 时会被认为是正确的。

Examples

stdin	stdout
1 1 1	2.2360679774998

Problem I. 叠塔游戏

Input file: **stdin**
Output file: **stdout**
Time limit: 20 seconds
Memory limit: 256 megabytes

小 Q 正在玩一个叠塔的游戏，游戏的目标是叠出尽可能高的塔。

在游戏中，一共有 n 张矩形卡片，其中第 i 张卡片的长度为 a_i ，宽度为 b_i 。小 Q 需要把所有卡片按一定顺序叠成一座塔，要求对于任意一个矩形，它的长度要严格大于它上边的任意一个矩形的长度。塔的高度为所有矩形的宽度之和。

在游戏中，小 Q 可以将卡片翻转 90 度来使用，而且必须用上全部 n 张卡片。

请写一个程序，帮助计算小 Q 能叠出最高的塔的高度。

Input

第一行包含一个正整数 $n(1 \leq n \leq 250000)$ ，即卡片的个数。

接下来 n 行，每行两个正整数 $a_i, b_i(1 \leq a_i, b_i \leq 10^9)$ ，分别表示每张卡片的长度和宽度。

Output

输出一行一个整数，即最高的塔的高度，输入数据保证一定存在解。

Examples

stdin	stdout
3 5 16 10 5 5 10	20