A. 期望逆序对

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

有 n 个独立的随机变量，其中 x_i 的值是一个从 $[l_i, r_i]$ 中随机选取的整数，即对于 $[l_i, r_i]$ 中的任何一个整数 j，$x_i = j$ 的概率都是 $(r_i - l_i + 1)^{-1}$。

现在你需要给出一个长度为 n 的排列 p，那么可以得到一个长度为 n 的随机变量序列 $x_{p1}, x_{p2}, \ldots, x_{pn}$。你的目标是让结果序列的逆序对个数的期望尽可能少。

求逆序对个数的期望的最小值。

Input

第一行输入一个整数 $n(1 \leq n \leq 5 \times 10^3)$。

接下来 n 行每行两个整数 $l_i, r_i(1 \leq l_i \leq r_i \leq 10^9)$。

Output

输出一行一个整数，表示答案对 998244353 取模后的值。假设答案的最简分数表示是 $\frac{x}{y}$，你需要输出一个整数 k 满足 $k \times y \equiv x \bmod 998244353$。

Example

input

```
3
1 2
2 3
1 3
```

output

```
332748118
```

B. 密码学

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

考虑一种加密方式，它需要一个任意长度的原文 m 和秘钥 *key*，其中要求原文和秘钥只包含大写和小写的英文字符。

首先定义字符之间的加密，用字符 a 去加密字符 b 的结果是：

1. 首先把 a 和 b 转成数字 x 和 y。转换的规则是，小写字母 a 到 z 依次对应 0 到 25，大写字母依次对应 26 到 51。
2. 计算 x 和 y 的和 z，对 52 取模，即计算 $(x + y) \bmod 52$。
3. 返回数字 z 对应的字符。

现在来讲如何用秘钥 *key* 来加密原文 m：

1. 如果秘钥的 *key* 的长度小于 m，那么不停重复 *key* 直到长度不小于 m 为止。举例来说，如果原文是 *beijing*，秘钥是 *PKUSAA*，那么秘钥需要被重复称 *PKUSAAPKUSAA*。
2. 假设原文的长度是 n，那么对于每一个 $[1, n]$ 的数字 i，都用 *key* 的第 i 个字符去加密 m 的第 i 个字符。
3. 返回结果。

那么用 *PKUSAA* 去加密 *beijing* 的结果就是：*QOcbINV*。

现在火山哥有 n 个字符串，s_1 到 s_n，他对这些字符串做了 m 次加密操作：第 i 次加密操作用第 s_{xi} 去加密 s_{yi}，并把 s_{y_i} 替换成加密结果。

现在依次给出 m 次加密操作，以及加密操作结束后每一个字符串的模样，你可以还原出这 n 个字符串原来的模样吗?

Input

第一行输入两个整数 $n, m (1 \leq n, m \leq 1000)$。

接下来 m 行每行输入两个整数 x_i, y_i，表示依次加密操作，保证 x_i 不等于 y_i。

接下来 n 行每行输入一个字符串，表示加密最后的结果。字符串的长度在 1 到 100 之间，只包含大小写英文字符。

Output

输出 n 行，每行一个字符串，表示原本的字符串。

Example

input

```
2 1
1 2
PKUSAA
QOcbINV
```

output

```
PKUSAA
beijing
```

C. 染色图

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

定义一张无向图 $G = \langle V, E \rangle$ 是 k 可染色的当且仅当存在函数 $f : V \mapsto \{1, 2, \ldots, k\}$ 满足对于 G 中的任何一条边 (u, v)，都有 $f(u) \neq f(v)$。

定义函数 $g(n, k)$ 的值为所有包含 n 个点的无自环、无重边的 k 可染色无向图中的边数最大值。举例来说，$g(3, 1) = 0, g(3, 2) = 2, g(3, 3) = 3$。

现在给出三个整数 n, l, r，你需要求解：

$$\left(\sum_{i=l}^{r} g(n,i)\right) \mod 998244353$$

Input

第一行输入一个整数 $T(1 \le T \le 10^3)$，表示数据组数。

对于每组数据，输入三个整数 $n,l,r(1 \le l \le r \le n \le 10^9)$。

Output

对于每组数据，输出一行一个整数表示答案。

Example

input
5 3 1 1 3 2 2 5 2 4 10 3 9 1000 123 789
output
0 2 23 280 332539617

D. 生成树

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

首先给出一些简单的概念：

- 对于一张 无向图 $G = \langle V, E \rangle$，树 $T = \langle V, E' \rangle$ 是 G 的生成树当且仅当 E' 是 E 的子集。
- 两棵 G 的生成树 $T_1 = \langle V, E_1 \rangle, T_2 = \langle V, E_2 \rangle$ 是不同的当且仅当它们使用的边集不同。
- 集合 $\mathcal{T}(G)$ 表示图 G 所有不同的生成树形成的集合。
- 函数 $s(G,T)$ 来衡量树 T 和图 G 的相似度，它的值等于同时出现在 T 和 G 中的边的数量。

现在给出两张 n 个点的无向图 G_1, G_2，你需要求：

$$\sum_{T \in \mathcal{T}(G_1)} s(G_2, T)$$

Input

第一行输入一个整数 $n(1 \le n \le 400)$ 表示点数。

接下来 n 行每行一个长度为 n 的 01 串，其中第 i 行第 j 位 $A_{i,j}$ 描述 G_1 中 i 到 j 是否有一条边。

接下来 n 行每行一个长度为 n 的 01 串，其中第 i 行第 j 位 $B_{i,j}$ 描述 G_2 中 i 到 j 是否有一条边。

输入保证：

- $\forall 1 \le i < j \le n, A_{i,j} = A_{j,i}, B_{i,j} = B_{j,i}$
- $\forall 1 \le i \le n, A_{i,i} = B_{i,i} = 0$
- $|\mathcal{T}(G_1)| \bmod 998244353 \ne 0$

Output

输出一行一个整数表示答案，答案可能很大，你只需要输出对 998244353 取模后的结果。

Example

input
3 011 101 110 010 101 010
output
4

E. 树与路径

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

在一棵有根树 T 上，任何两点间的最短路径都能够分为两个阶段：

1. 从起点出发，沿着向根的方向走若干条边。
2. 向着终点，沿着离开根的方向走若干条边。

定义一条路径的权值为向上走的边数乘上向下走的边数。特殊地，当起点等于终点的时候，两阶段的边数都是 0；当起点是终点的祖先的时候，第一阶段的边数是 0；当终点是起点的祖先的时候，第二阶段的边数是 0——这三种情况下，路径的权值都是 0。

现在给出一棵 n 个节点的无根树 T 和 m 条路径 (a_i, b_i)。对于每一个 $r \in [1, n]$，你需要计算当 r 是根节点的时候，所有路径的权值和是多少。

Input

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

接下来 $n-1$ 行每行输入两个整数 $u_i, v_i (1 \le u_i, v_i \le n)$，表示树上的一条边。

接下来 m 行每行输入两个整数 $a_i, b_i (1 \le a_i, b_i \le n)$，表示一条路径。

Output

输出 n 行每行一个整数，第 i 行表示以 i 为根时，所有路径的权值和。

Example

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

F. 乘法

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

给出一个长度为 n 的数列 $A_1, \ldots, A_n$ 和一个长度为 m 的数列 $B_1, \ldots, B_m$，可以构造得到一个 $n \times m$ 的矩阵 C，其中 $C_{i,j} = A_i \times B_j$。

给出整数 K，你需要求出 C 中第 K 大的数的值。

Input

第一行输入三个整数 $n, m, K (1 \le n, m \le 10^5, 1 \le K \le n \times m)$。

第二行输入 n 个空格隔开的整数 $A_1, \ldots, A_n (-10^6 \le A_i \le 10^6)$。

第三行输入 m 个空格隔开的整数 $B_1, \ldots, B_m (-10^6 \le B_i \le 10^6)$。

Output

输出一行一个整数，表示矩阵中的第 K 大的数的值。

Example

input
3 3 3 2 3 4 4 5 6
output
18

G. 圆凸包

time limit per test: 1 second

memory limit per test: 256 megabytes
input: standard input
output: standard output

给出 n 个点求他们的凸包是一个经典问题，所以出了一道稍微难一点的题。

给出平面上 n 个圆，第 i 个圆的圆心是 (x_i, y_i)，半径是 r_i。 定义这 n 个点的凸包为所有满足以下条件的点 P 形成的区域：存在点 A, B 和常数 $\alpha \in [0, 1]$ 满足 A, B 都在某个圆的内部（所在的圆可以不同）且 $P = \alpha A + (1 - \alpha) B$。换句话说，这 n 个点的凸包等于这 n 个圆内部的所有点形成的凸包。

现在给出这 n 个圆，试求这 n 个圆形成的凸包的周长。

Input

第一行输入一个整数 $t (1 \le t \le 20)$ 表示数据组数。

每组数据的第一行是一个整数 $n (1 \le n \le 100)$ 表示圆的个数。

接下来 n 行每行三个整数 $x_i, y_i, r_i (1 \le r_i \le 10^3, |x_i|, |y_i| \le 10^3)$，描述了一个圆。

Output

对每组数据输出一行一个实数，表示周长。你的答案会被视为正确当且仅当相对误差或者绝对误差不超过 10^{-6}。

Example

input

```
3
2
0 0 1
1 0 1
4
0 0 1
0 1 1
1 0 1
1 1 1
5
0 0 1
2 2 1
0 2 1
2 0 1
1 1 2
```

output

```
8.28318530718
10.28318530718
14.28318530718
```

H. 最大公约数

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

有三个人，A, B, C，其中 A 和 B 共享了一个神秘的数字 k，已知 $1 \le k \le n$。

现在 A 和 C 说："k 的值等于 x"。

C 不太信任 A，于是想向 B 确认一下 k 是否真的等于 x。B 虽然不想直接把 k 的值告诉 C，但是 B 允许 C 给出一个正整数 y（注意 y 可以大于 n），然后 B 会回答 $\gcd(k, y)$。

现在给出 k, n，你需要帮助 C 决定这样的 y 的取值，使得 C 一定可以通过 B 的回答来判断 A 有没有撒谎。如果这样的 y 有多个，你需要输出最小的那个。

Input

输入第一行是一个整数 $T(1 \leq T \leq 50)$。

对于每组数据，输入一行两个整数 $n, k(1 \leq k \leq n \leq 500)$。

Output

对于每组数据，输出一行一个整数，表示答案。如果满足条件的 y 不存在，则输出 -1。

Example

input
3 10 1 10 4 10 7
output
210 8 7

I. K 小数查询

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

热爱学习刻苦奋斗的九条可怜最近做了很多数据结构题，接触到了 K 小数查询这一系列的问题以及线段树的重磅打击这一套理论，她觉得这两样东西都很厉害，所以想要出一道题。

给出一个长度为 n 的数列 A，接下来有 m 次操作，操作有两种：

- $1\ l\ r\ x$，表示对 $i \in [l, r]$，令 $A_i = \min(A_i, x)$
- $2\ l\ r\ k$，表示询问区间 $[l, r]$ 中第 k 小的数。

这个问题对可怜来说有点难，你能帮帮她吗。

Input

第一行输入两个整数 $n, m(1 \leq n, m \leq 8 \times 10^4)$。

接下来一行 n 个整数描述数组 $A(1 \leq A_i \leq n)$。

接下来 m 行每行描述一个操作，操作格式与题面中相同，保证 $1 \leq l \leq r \leq n, 1 \leq k \leq r - l + 1, 1 \leq x \leq 10^9$。

Output

对于每组询问，输出一个整数表示答案。

Example

input
3 5 1 2 3 2 1 3 2 1 3 3 1 2 1 3 2 1 1 2 3 2 1 3 2
output
2 1 1

J. 德州扑克

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

最近沉迷德州扑克，于是就出了一道有关德州扑克的题。

一副去掉大小王的扑克牌包含 13 种不同的数值：$2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A$，它们的大小从左到右依次递增。扑克牌中有四种不同的花色，用 $0, 1, 2, 3$ 表示，每一种花色都有 13 张牌，分别对应每一种数值。在德州扑克中，我们只需要考虑这 52 张牌。

一副手牌包含 5 张扑克牌，他们可能会形成若干种牌型，按照从大到小的顺序依次为：

1. 同花顺：花色相同的顺子（顺子的定义见下方），例如同花色的 T, J, Q, K, A。
2. 四条：存在四张大小相同的牌，例如任意花色的 $T, T, T, T, 2$。
3. 葫芦：有三张牌大小相同，另外两张牌大小相同，例如任意花色的 T, T, T, J, J。
4. 同花：五张牌花色相同，例如同花色的 $7, J, Q, K, A$。
5. 顺子：五张牌大小连续，例如任意花色的 $2, 3, 4, 5, 6$, T, J, Q, K, A。特殊地，$A, 2, 3, 4, 5$ 也是一个顺子（但是 $K, A, 2, 3, 4$ 不是）。因此一共有 10 种不同数值的顺子，它们的第一张牌分别是 $A, 2, 3, 4, 5, 6, 7, 8, 9, T$。
6. 三条：存在三张大小相同的牌，例如任意花色的 T, T, T, J, Q。
7. 两对：存在两个大小不同的对子（一个对子是两张大小一样的牌），例如任意花色的 T, T, Q, Q, K。
8. 对子：存在两张大小相同的牌，例如任意花色的 T, T, J, Q, K。
9. 高牌：不满足以上任何一个牌型的手牌都是高牌。

一副手牌可能同时满足很多个不同的牌型，这个时候我们会把最大的那个牌型作为这幅手牌的牌型。

我们可以通过如下方式来比较两幅手牌的大小：

- 如果两幅手牌牌型不同，那么牌型较大的手牌更大。

- 如果两幅手牌都是顺子或者都是同花顺，那么按照顺子第一张牌的大小排序，从小到大分别为 $A, 2, 3, 4, 5, 6, 7, 8, 9, T$，即 $A, 2, 3, 4, 5$ 是最小的顺子，T, J, Q, K, A 是最大的顺子。
- 否则，我们按照（出现次数，数值）的双关键字从大到小把这五张牌排序，例如 K, K, T, T, T 排序后就是 T, T, T, K, K；$2, T, T, K, A$ 排序后是 $T, T, A, K, 2$。
- 比较两幅牌的字典序。

注意，牌的花色只影响第一步比较牌型，并不影响后面两步的比大小。

下面是一些例子：

- 相同花色的 $5, 6, 7, 8, 9$ 大于相同花色的 $A, 2, 3, 4, 5$。
- 相同花色的 $A, 2, 3, 4, 5$ 大于相同花色的 $2, 4, 5, 6, 7$。
- 任意花色的 $3, 3, 8, 8, K$ 大于任意花色的 $5, 5, 7, 7, A$。
- 任意花色的 Q, Q, Q, T, T 大于任意花色的 J, J, J, A, A。

在德州扑克中，一个人的场面包含 7 张扑克牌，这 7 张牌可以形成 $\binom{7}{5}$ 种不同的手牌，而这个人场面的大小等于这些手牌中最大的那一个。

现在两个人单挑：

- 首先这两个人分别从一副（去掉大小王）的扑克牌中抽了 2 张牌。
- 剩下的 48 张牌被完全均匀地打乱（即 $48!$ 种顺序等概率出现）。
- 最前面的 5 张牌公开被公开。
- 每个人的场面由自己的 2 张牌加上公开的 5 张牌组成，两个人场面较大的那一方赢。

注意平局是可能发生的，例如两个人的手牌分别是花色 0 的 $2, 3$ 和花色 1 的 $2, 3$，公共牌是花色 2 的 T, J, Q, K, A，那么两个人的场面都是 T, J, Q, K, A 的同花顺，因此大小相同。

现在给出两个人的初始手牌，你需要计算两个人每个人获胜的概率以及平局的概率。

Input

第一行输入一个整数 $T(1 \leq T \leq 10)$ 表示数据组数。

对于每组数据，输入包括四行，其中每行包含两个整数 $c_i, w_i(c_i \in \{0, 1, 2, 3\}, w_i \in \{2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A\})$，描述了一张牌。其中前两张是第一个人的手牌，后两张是第二个人的手牌。

输入保证这四张牌两两不同。

Output

对于每组数据输出三行，分别表示第一个人获胜的概率，平局的概率和第二个人获胜的概率。概率以最简分数的形式输出：例如 0.5 就输出 $1/2$；0 就输出 $0/1$；0.999 就输出 $999/1000$。

Example

input

```
3
0 A
1 A
2 A
3 A
0 2
1 2
```

```
2 2
3 2
0 A
1 Q
2 K
2 J
```

output

```
18605/856152
409471/428076
18605/856152
8917/428076
205121/214038
8917/428076
1000357/1712304
1891/428076
704383/1712304
```