Problem A. 字符串大师 II

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

对于一个字符串 $S[1..n]$ 来说，定义 $border[i] = \max\{k|0 \le k < i, S[1..k] = S[i-k+1..i]\}$。比如对于 $S =$`abacabaa`，$border[1..8] = [0,0,1,0,1,2,3,1]$。

定义字符集为 60 的数字串 $G_1 = 1, G_k = G_{k-1} + k + G_{k-1}(k \ge 2)$，其中“+”表示字符串拼接，当 $k \ge 10$ 时将其作为一个整体而不是两个字符。例如：$G_4 =$`1 2 1 3 1 2 1 4 1 2 1 3 1 2 1`。

给定 k 和 p，你的任务是对于 $S = G_k$，求出 $border[p]$ 的值。

Input

第一行包含一个正整数 $T(1 \le T \le 100000)$，表示测试数据的组数。

每组数据包含一行两个正整数 $k, p(1 \le k \le 60, 1 \le p < 2^k)$。

Output

对于每组数据输出一行一个整数，即 $border[p]$ 的值。

Examples

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

Problem B. 超速摄像头

Input file:	`stdin`
Output file:	`stdout`
Time limit:	15 seconds
Memory limit:	512 megabytes

比特镇一共有 n 个路口，编号依次为 1 到 n。这些路口之间通过 $n-1$ 条双向道路连接成了一棵树的结构。任意两点间都有且仅有一条直接或间接到达的路径。

为了建立更好的交通秩序，比特镇计划在某些路口安装超速摄像头，每个路口最多只能安装一个摄像头。同时为了不让人们的神经过于紧张，对于任意两点 u, v 的最短路径 (包括 u 和 v 两点)，路径上安装的摄像头总数不能超过 k。

摄像头厂商小 Q 希望在满足以上限制的情况下，在比特镇安装尽可能多的超速摄像头。请写一个程序，帮助小 Q 找到最佳的安装方案。

Input

第一行包含两个正整数 $n, k (1 \le n, k \le 1000000)$，分别表示点数和上限。

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

Output

输出一行一个整数，即最多能安装的超速摄像头的数量。

Examples

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

Problem C. 质数拆分

Input file:	`stdin`
Output file:	`stdout`
Time limit:	4 seconds
Memory limit:	512 megabytes

给定一个正整数 n，求方程 $a+b+c+d=n$ 的解的个数，要求 a,b,c,d 都是质数，其中 a,b,c,d 可以相等。

Input

第一行包含一个正整数 $n(1 \le n \le 150000)$。

Output

输出一行一个整数，即解的个数。

Examples

stdin	stdout
9	4

Problem D. 航海计划

Input file:	`stdin`
Output file:	`stdout`
Time limit:	15 seconds
Memory limit:	512 megabytes

在比特海上一共有 n 座大陆，编号依次为 1 到 n。航海家小 Q 正位于 1 号大陆上，他计划航行去 n 号大陆。

这些大陆之间通过 m 条单向航线连接，每个大陆最多只会作为 10 条单向航线的起点。第 i 条航线起点为 a_i，终点为 b_i，耗费 d_i 天。这意味着如果在第 j 天从 a_i 号大陆出发，那么会在第 $j+d_i$ 天到达 b_i 号大陆。小 Q 可以在任何大陆留宿任意长的时间。但是，一旦选择了某条航线，中途不能取消，也不能改变路线和到达时间。小 Q 也不能选择不存在的航线，因为开辟新航线非常危险。在一开始 (第 1 天)，小 Q 位于 1 号大陆。

每座大陆都可能会有一些不太平的时候。具体来说，一共有 p 段区间，第 i 段区间发生在 w_i 号大陆上，在第 s_i 到第 k_i 天 (包括两端点) 都不安全。任何时刻，小 Q 都不能位于一座不安全的大陆上。不过幸运的是，在第 1 天，1 号大陆是安全的。

请帮助小 Q 找到一个最佳方案，使得他能尽早到达 n 号大陆。

Input

第一行包含两个正整数 $n, m(2 \le n \le 100000, 1 \le m \le 1000000)$，分别表示大陆的数量和航线的数量。

接下来 m 行，每行三个正整数 $a_i, b_i, d_i(1 \le a_i, b_i \le n, a_i \ne b_i, 1 \le d_i \le 10^9)$，表示一条单向航线。

接下来一行包含一个整数 $p(0 \le p \le 100000)$，表示不安全的区间数量。

接下来 p 行，每行三个正整数 $w_i, s_i, k_i(1 \le w_i < n, 1 \le s_i \le k_i \le 10^9)$，表示一段不安全区间。同一个大陆上不同区间之间可能有重叠部分。

数据保证在第 1 天，1 号大陆是安全的，且每个大陆最多只会作为 10 条单向航线的起点。

Output

若找不到可行方案，输出“`-1`”，否则输出一个整数 d，表示最佳方案中在第 $d+1$ 天到达 n 号大陆。

Examples

stdin	stdout
5 6 1 2 3 1 4 13 2 3 1 2 4 2 3 2 2 4 5 1 5 1 2 4 1 8 8 2 6 7 2 10 11 4 6 7	10

Problem E. 比例查询

Input file:	`stdin`
Output file:	`stdout`
Time limit:	8 seconds
Memory limit:	512 megabytes

给定一个长度为 n 的正整数序列 $a_1, a_2, ..., a_n$，m 次询问 l_i, r_i, b_i。

请对于每个询问，判断是否存在两个位置 $x, y(l_i \le x, y \le r_i, x \ne y)$，满足 $\frac{a_x}{a_y} = b_i$。

Input

第一行包含两个正整数 $n, m(2 \le n, m \le 100000)$，分别表示序列长度以及询问个数。

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

接下来 m 行，每行三个正整数 $l_i, r_i, b_i(1 \le l_i < r_i \le n, 2 \le b_i \le 100000)$，依次表示每个询问。

Output

对于每个询问输出一行，若存在满足条件的两个位置，输出“`Yes`”，否则输出“`No`”。

Examples

stdin	stdout
4 2 5 8 3 9 3 4 3 1 4 6	Yes No

Problem F. 糖果商店

Input file:	`stdin`
Output file:	`stdout`
Time limit:	15 seconds
Memory limit:	512 megabytes

小 Q 知道小 T 非常喜欢吃糖果，准备购买一些糖果作为礼物送给小 T。

在糖果商店里一共有 n 种糖果，编号依次为 1 到 n，第 i 种糖果的重量为 w_i，美味度为 v_i，每种糖果的数量都是无限的。

每次小 Q 可以选定一种糖果 i，然后选定一个整数 j，将 j 颗第 i 种糖果打包成一盒。为了防止盒子过空，商店规定 j 至少为 l_i。

小 Q 的礼物包含若干盒种类可能不同的糖果，重量为所有糖果的重量之和，价值是所有糖果的美味度之和。他把这些盒子按照一定顺序从下到上叠起来，每次叠上一盒糖果时，这盒糖果会压住下方 (不包括本身) 的所有糖果，假设下方有总重量为 $sum(sum \geq 1)$ 的糖果，则会使礼物的最终价值减少 d_{sum}。

请写一个程序，帮助小 Q 准备出价值最高的礼物。当然，小 Q 可以不买任何糖果，此时价值为 0。

Input

第一行包含两个正整数 $n, m(1 \leq n \leq 100, 1 \leq m \leq 100000)$，表示糖果的种类数以及总重量的上限。

第二行包含 m 个整数 $d_1, d_2, ..., d_m(0 \leq d_i \leq 10^9)$，表示叠糖果对礼物价值的损耗。

接下来 n 行，每行三个正整数 $w_i, v_i, l_i(1 \leq w_i, l_i \leq m, 1 \leq v_i \leq 10^9)$，依次描述每种糖果。

Output

输出一行 m 个整数，其中第 i 个整数表示礼物总重量不超过 i 时价值的最大可能值。

Examples

stdin	stdout
3 6 1 0 0 0 2 0 5 30 1 1 4 4 1 3 1	3 6 9 16 30 32

Problem G. 最长公共子序列

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

一个字符串 S 的子序列是指从 S 中删除若干项 (可以不删，也可以全删完)，然后将剩下的字符按照原顺序从左往右顺次连接得到的字符串。

两个字符串 S 和 T 的最长公共子序列是指最长的字符串 P，满足 P 既是 S 的子序列，又是 T 的子序列。

给定一个长度为 n 的小写字母构成的字符串 S，请找到一个长度同样为 n 的小写字母构成的字符串 T，使得 S 和 T 的最长公共子序列最短。

Input

包含一行一个小写字母构成的字符串 $S(1 \leq |S| \leq 1000000)$。

Output

输出一行一个整数，即你找到的 T 与 S 的最长公共子序列的长度。

Examples

stdin	stdout
abc	0

Problem H. 路径统计

Input file:	`stdin`
Output file:	`stdout`
Time limit:	35 seconds
Memory limit:	512 megabytes

给定一棵 n 个点的树，编号依次为 1 到 n，任意两点之间有且仅有一条直接或间接路径。

令 $S(u,v)$ 表示在树上 u 到 v 的唯一最短路径上，所有点 (包括 u 和 v) 的编号组成的集合。

请写一个程序，统计有多少个正整数对 (u,v) 满足 $1 \le u \le v \le n$，且将 $S(u,v)$ 从小到大排序后，相邻两个数的差值恰好为 1。

Input

第一行包含一个正整数 $n(1 \le n \le 250000)$，表示点数。

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

Output

输出一行一个整数，即满足条件的点对 (u,v) 的数量。

Examples

stdin	stdout
3 1 2 1 3	5

Problem I. 最远点对

Input file:	`stdin`
Output file:	`stdout`
Time limit:	25 seconds
Memory limit:	512 megabytes

在平面上有 n 个点，编号依次为 1 到 n，第 i 个点坐标为 (x_i, y_i)。

m 次询问，每次给定两个正整数 $l_i, r_i (1 \le l_i < r_i \le n)$，请找到两个点 $a, b (l_i \le a, b \le r_i)$，使得它们的欧几里得距离最大。

Input

第一行包含两个正整数 $n, m (2 \le n, m \le 120000)$，分别表示点数和询问数。

接下来 n 行，每行两个正整数 $x_i, y_i (1 \le x_i, y_i \le 10^9)$，分别表示每个点的坐标。不同点的坐标可能相同。数据保证所有 x_i 和 y_i 都是在 $[1, 10^9]$ 里等概率随机选取的。

接下来 m 行，每行两个正整数 $l_i, r_i (1 \le l_i < r_i \le n)$，依次表示每个询问。

Output

对于每个询问输出一行一个整数，即你找到的点对的距离的平方。

Examples

stdin	stdout
5 3 1 3 2 1 4 5 5 2 3 5 1 5 2 3 3 5	20 20 13