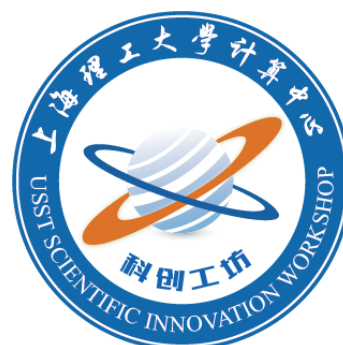


USSTSIW-ACM 集训队选拔赛

June 7, 2019



上海理工大学
UNIVERSITY OF SHANGHAI FOR SCIENCE AND TECHNOLOGY



Problems

- A Haruhi with Acn-hcpc
- B Haruhi with Bubbles
- C Haruhi with Changes
- D Haruhi with "Day of sagittarius"
- E Haruhi with Equilibrium game
- F Haruhi with Fish
- G Haruhi with Global offensive
- H Haruhi with Haruhi with Haruhi
- I Haruhi with Inconsistent rain
- J Haruhi with Jenga
- K Haruhi with K-intimacy
- L Haruhi with Literature club
- M Haruhi with Mogician

比赛开始前请勿提前翻阅！

比赛注意事项

技术相关：

1. *C++* 版本为 *C++14*，开 *-O2* 优化。
2. *Java* 允许 *package* 语句与任意类名。
3. *CompileError* 不计罚时，可点击提交条目查看编译器输出。
4. 源代码体积限制为 256kB 。

比赛规则相关：

1. 最后一小时封榜，所有提交在榜单上显示为 *Pending*（评测正常进行），不发气球/旗子。
2. 使用页面上的 *request clarification* 发起提问，例如向裁判组请求对题意/样例的解释，以及其他问题。裁判组的回复或全体通知会出现在页面右侧 *Clarification* 栏，加粗为未读，请及时查看。
3. 提交代码时点击右上角 *submit* 提交源文件(.c/.cpp/.java/.py)，使用恰当的文件名与后缀可以自动匹配题目与语言，例如 A.cpp 会识别为 A 题的 *C++* 提交。

Problem A. Haruhi with Acm-hcpc

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

兵庫県立西宮北高等学校马上就要举办第一届 *ACM-HCPC* 了, *Haruhi* 已经迫不及待想要拿冠军了, 现在“邀请”你编写一份滚榜程序。

注意你的滚榜程序必须满足以下要求, 不然你会被 *Haruhi* ...。

1. 组委会会提供输入文件, 文件提供了封榜前的榜, 和封榜后的提交信息。格式详见输入格式。

2. 关于排名:

- i. 解出题目数多的队伍排名较前;
- ii. 解出题目数相同, 则总罚时较少的队伍排名较前;
- iii. 解出题目数和总罚时都相同, 则最后解出题目的时间更早的队伍排名较前。

3. 关于罚时: 第 i 题对罚时的贡献 $T_i = 20 \times$ 第一次 *AC* 前错误的次数 + 第一次 *AC* 的时间。总罚时 $= \sum T_i$ 。也就是说, 只有在第一次通过该题时才结算罚时。

由于糟糕的评测平台, 提交代码的结果只有 *WA(Wrong Answer)* 和 *AC(Accepted)* 两种。

Haruhi 对谁过了哪些题不感兴趣, 所以你只需要按排名输出队名、过题数以及罚时, 输入保证没有并列。

Input

第一行两个正整数 $n, m (1 \leq n \leq 1000, 1 \leq m \leq 15)$, 分别表示参赛队伍的数量和题目的数量。

接下来 $3 \times n$ 行。

其中的第 $3 \times k - 2$ 行一个字符串 $s (1 \leq |s| \leq 100)$ 表示第 k 支队伍的名字, 只包含可见字符。

第 $3 \times k - 1$ 行有 m 个字符串, 表示第 k 支队题目的提交信息, 格式如下。

- $+p_{k,i}$ 表示 *WA* 了 $p_{k,i}$ 次后 *AC* 了第 i 题。特别地, 单独一个 $+$ 表示第一次提交就 *AC*。
- $-p_{k,i}$ 表示 *WA* 了 $p_{k,i}$ 次后仍未 *AC* 第 i 题。
- 0 表示未曾提交这题。
- 其中 $1 \leq p_{k,i} \leq 100$ 。

第 $3 \times k$ 行有 m 个整数 $q_{k,i} (-1 \leq q_{k,i} < 240)$, 表示第 k 支队第一次 *AC* 了第 i 题时的提交时间。数据保证 $q_{k,i} = -1$ 当且仅当该队没有通过此题。

再接下来一行一个整数 $h (0 \leq h \leq 1000)$, 表示封榜后有 h 次提交。

接下来 h 行, 每行四个正整数 $x, y, z, t (1 \leq x \leq n, 1 \leq y \leq m, z \in \{0, 1\}, 240 \leq t < 300)$, 表示第 x 支队伍在第 t 分钟提交了 y 题评测结果为 z , 其中 $z = 1$ 表示 *AC*, $z = 0$ 表示 *WA*。

Output

按排名输出 n 行, 每一行包含一个字符串 s 和两个正整数 $solved, penalty$, 用一个空格分隔开, 分别表

示队伍的名字、过题数和罚时。

Samples

standard input	standard output
5 4 SOS_Dan -1 -2 0 + -1 -1 -1 30 jxcyxctxc -4 -5 -6 -7 -1 -1 -1 -1 I_AK_IOI + + + + 10 110 70 30 Sasaki -1 -1 -1 -1 -1 -1 -1 -1 Pony.AI + + + + 50 60 70 40 10 1 1 1 250 1 2 1 255 1 3 0 256 1 3 1 257 1 3 1 260 2 1 1 244 2 2 1 299 3 1 1 240 4 1 0 241 4 2 0 242	Pony.AI 4 220 I_AK_IOI 4 220 SOS_Dan 4 872 jxcyxctxc 2 723 Sasaki 0 0

Problem B. Haruhi with Bubbles

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

Haruhi 在社团教室里养了两条鱼，一条叫红鲤鱼，一条叫绿鲤鱼。

Haruhi 发现红鲤鱼和绿鲤鱼喜欢吐泡泡。

吐出的泡泡可以看成一串仅含有 O 和 o 的字符串。

每过一秒，两个连续的 o 会变成一个 O ，两个连续的 O 会爆掉。

Haruhi 没有那么多时间看鱼吐泡泡，现在有一串初始的泡泡序列，她想知道最终的泡泡序列。

Input

输入只有一行，是一个仅含有 O 和 o 的字符串 s ($1 \leq |s| \leq 1000$)。

Output

输出一行字符串，表示泡泡序列的最终形态。

特别地，如果所有泡泡都爆掉了，输出 0。(这是个零)

Samples

standard input	standard output
oo00oooo0	o0

Note

泡泡们的合并是从左往右哒。

Problem C. Haruhi with Changes

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

有一个 $n * m$ 的全 1 矩阵 S ，这里定义一种操作可以使得 S_{ij} 及其 S_{ij} 四联通区域内的元素（ S_{ij} 表示矩阵 S 中第 i 行第 j 列的元素）从 1 变成 0，或者从 0 变成 1。

已知 Haruhi 对矩阵 S 中的每一个元素 S_{ij} 都有对应的若干次操作 a_{ij} 。

现在，请你输出所有的变化操作之后的 S 矩阵。

Input

第一行输入两个正整数 n, m ，表示 S 矩阵的行数和列数 ($1 \leq n, m \leq 1000$)。

接下来输入一个 $n * m$ 行的二维矩阵 A ， A 矩阵中的每一个元素 a_{ij} ($0 \leq a_{ij} \leq 10^6$) 对应 S 矩阵中每一个元素 S_{ij} 被操作的次数。

Output

输出所有的变化操作之后的 S 矩阵(注意样例的输出格式，每一行的行末不得有多余的空格)。

Samples

standard input	standard output
3 3 1 0 0 0 0 0 0 0 1	001 010 100
3 3 1 0 1 8 8 8 2 0 3	010 011 100

Problem D. Haruhi with "Day of sagittarius"

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

Haruhi 舰长发现敌军掌控着 n 个星球，这 n 个星球之间有 m 条双向航道(不保证联通，可能有自环和重边)。

为了切断敌军的补给线，*Haruhi* 舰长能够在某些星球放置她的战舰。当 x 星球被 *Haruhi* 放置的军舰占领时，所有通向 x 星球或从 x 星球出发的航道将被封锁而不能再使用。她想要知道被切断航道后，未被占领的星球间组成的联通块有几个。

对于她的每次放置，你都要回答一次。

Input

第一行两个正整数 $n, m (1 \leq n \leq 2 * 10^5, 1 \leq m \leq 2 * 10^5)$ ，表示有 n 个星球(从 1 开始编号)和 m 条航道。

接下来 m 行，每行两个正整数 $u_i, v_i (1 \leq u_i, v_i \leq n)$ ，表示 u_i 星球和 v_i 星球之间有一条航道。

然后一行 $k (1 \leq k \leq n)$ ，表示 *Haruhi* 想要放置 k 艘战舰。

接下来 k 行，每行一个正整数 $x_i (1 \leq x_i \leq n)$ ，表示 *Haruhi* 想要在 x_i 星球放置战舰。

输入保证所有的 x_i 都互不相同。

Output

总共 k 行，每行一个正整数，表示星球间的联通块数。

Sample

standard input	standard output
4 4	1
1 2	2
1 3	1
3 4	0
2 4	
4	
2	
3	
1	
4	

Problem E. Haruhi with Equilibrium game

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

Haruhi 迷上了新的游戏，她想和你比一比。

n 堆石子围成一个圈，玩 n 轮游戏，每轮独立。第 i 轮从第 i 堆开始。

游戏规则是：从起始堆开始，两人轮流取石子，每次可以拿任意多个，最少拿 1 个，最多只能把当前堆取完，当前堆拿完之后才能进行下一堆，直到全部 n 堆取完，谁不能取谁就输啦。

当然你必须要让 *Haruhi* 赢，不然我们可能要在下一个世界才能看到你啦。为此你必须判断某轮中先手是否有必胜策略。

注意当你选定先手或后手后，你和 *Haruhi* 只能按照各自的最优策略取石子，不然 *Haruhi* 会认为你在让她。

Input

第一行 $n(1 \leq n \leq 10^5)$ 。

第二行 n 个整数 $a_i(1 \leq a_i \leq 10^9)$ ，表示第 i 堆石头的数量。

Output

输出 n 行，第 i 行表示第 i 轮游戏先手是否有必胜策略。

如果先手有必胜策略，输出“Haruhi go first.”(不含双引号)，

反之输出“Haruhi go secondary.”(不含双引号)。

Sample

standard input	standard output
3	Haruhi go first.
2 1 3	Haruhi go secondary.
	Haruhi go first.

Problem F. Haruhi with Fish

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

Haruhi 在社团教室里养了两条鱼，一条叫红鲤鱼，一条叫绿鲤鱼。

Haruhi 发现红鲤鱼和绿鲤鱼喜欢吐泡泡。

吐出的泡泡可以看成一串仅含有 O 和 o 的字符串。

红鲤鱼吐出的泡泡一定包含 OoO 或者 OOO 这样的连续子串。

绿鲤鱼吐出的泡泡一定不包含 OoO 和 OOO 这样的连续子串。

Haruhi 想知道所有长度为 n 的泡泡串中，可能是绿鲤鱼吐出的方案有多少种。

Input

一个正整数 $n(1 \leq n \leq 10^6)$ 表示泡泡串长度。

Output

一个整数表示答案，由于答案可能很大，请对 99037660 取模。

Sample

standard input	standard output
3	6

Problem G. Haruhi with Global offensive

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

夏天到了, *SOS* 团要进行水枪大战了, 我们的 *Haruhi* 可是宇宙第一不想输的。

Haruhi 为此练习起了枪法。已知她的水枪的有效射程是 $10m$, 有效范围是 r 。为了方便起见, 我们取 *Haruhi* 面对的 $10m$ 外的竖直平面建立笛卡尔平面直角坐标系。我们认为当她瞄准着 (x_1, y_1) 这个点时, 她的子弹会等概率的出现以 (x_1, y_1) 为圆心半径为 r 的圆内。现在, 平面上有一块以 (x_2, y_2) 为圆心 R 为半径的圆形靶子。 *Haruhi* 想知道她瞄准着 (x_1, y_1) 时, 有多大的概率能打中靶子。

Input

多组数据。

第一行一个正整数 $T(1 \leq T \leq 100)$, 表示有 T 组数据。

接下来 T 行, 每行六个整数 $x_1, y_1, r, x_2, y_2, R(-100 \leq x_1, y_1, x_2, y_2 \leq 100, 1 \leq r, R \leq 400)$ 。

Output

对于每组数据, 输出一行一个小数, 表示击中的概率, 四舍五入到小数点后第四位。

Sample

standard input	standard output
4	0.0000
0 0 1 0 2 1	1.0000
0 0 1 0 0 2	0.2500
0 0 2 0 0 1	0.1817
0 0 1 1 1 1	

Note

$\pi = 3.14159265358979\dots$

C/CPP 选手请不要使用 *y1* 作为你的变量名, 不然你可能会得到一个 *CompileError*。

Problem H. Haruhi with Haruhi with Haruhi

Input file: standard input

Time limit: 1.5 seconds

Output file: standard output

Memory limit: 1024 megabytes

实力，心态，策略，临场甚至运气，缺一不可^[1]。

Haruhi 梦到了自己在报名参加 *ACM - HCPC*，她需要挑选好几个队友。但是她惊然地发现所有人都变成了 *haruhi*，这些 *haruhi* 都是五边形战士，也就是说她们拥有着实力、心态、策略、临场、运气这五个维度的能力。为了方便，*Haruhi* 测定了每一个 *haruhi* 的能力，对于每一个维度，给出她们在 n 个 *haruhi* 中的排名，分别用 a_i, b_i, c_i, d_i, e_i 表示，每个维度的排名都不会有并列。

现在 *Haruhi* 想要知道对于每一个 *haruhi*，有多少个 *haruhi* 的能力全面碾压了她(即五个能力均比她排名高)。

Input

第一行一个正整数 $n(1 \leq n \leq 30000)$ 。

接下来 n 行，每行 5 个正整数 $a_i, b_i, c_i, d_i, e_i(1 \leq a_i, b_i, c_i, d_i, e_i \leq n)$ ，分别表示第 i 个 *haruhi* 五维能力的排名。

Output

输出 n 行，第 i 行表示第 i 个 *haruhi* 被多少人全面碾压。

Sample

standard input	standard output
4	0
1 2 3 1 1	0
2 1 1 2 2	1
3 3 2 3 3	3
4 4 4 4 4	

Note

如果没有非常自信的低复杂度解法，建议使用 *C/C++* 编写此题。

[1] <https://www.zhihu.com/question/324603385/answer/688828615>

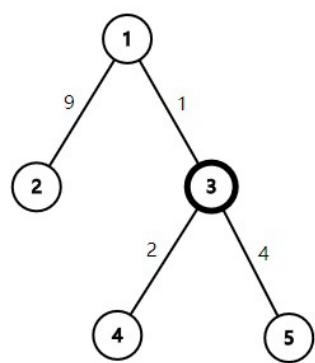
Problem I. Haruhi with Inconsistent rain

Input file: standard input Time limit: 1 second
Output file: standard output Memory limit: 256 megabytes

熟悉而又平凡的日常。

Haruhi 不知道又用了什么方法获取到了电器街老板们的赞助，电器街上的每一家都答应赠送 Haruhi 若干电器，还可以选择一家店免费租用他们的仓库。而搬运电器这个光荣的使命自然而然地就落到了你的身上。

你需要选择一家店作为仓库，任务是把所有的电器都搬回仓库，遗憾的是由于电器实在是太沉了，每一次只能搬一台电器。已知电器街所有店之间的道路恰好构成一棵无根树(如下图)。那么现在任务就变得很简单了，选择哪家电器作为仓库可以使搬完所有电器的路程最短呢。



Input

第一行一个正整数 $n(2 \leq n \leq 10^5)$ ，表示电器街上店的数量。

接下来一行 n 个整数 $e_i(1 \leq e_i \leq 10^6)$ ，表示店 i 要赠送给 Haruhi e_i 台电器。

接下来 $n - 1$ 行，每行包含三个正整数 $x, y, v(1 \leq x, y \leq n, 1 \leq v \leq 10^5)$ ，表示店 x 与店 y 之间有一条长为 v 的道路。

Output

第一行一个正整数 X ，表示选择 X 店作为仓库时路程总和最短 (有多个答案时请输出 X 最小的那一个)。

第二行一个整数 S ，表示此时的总路程。

Sample

standard input	standard output
5	3
1 2 3 4 5	98
1 2 9	
1 3 1	
3 4 2	
3 5 4	

Problem J. Haruhi with Jenga

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

电脑社社长要和 *Haruhi* 决战叠叠乐大赛!

电研社社长的要求如下:

- 用电研社自创的叠叠乐游戏一决胜负。
- 假如我们赢了, 目前安置在 *SOS* 团桌上的电脑, 就要乖乖回到原本应在的场所。
- 归根究底, *SOS* 团根本就用不到那么好的多功能型电脑, 电脑应该是放在电脑研究社才能物尽其用的器材。强烈要求物归原主。
- 抢夺电脑时带给社长和社员们精神上莫大的苦痛, 就趁这机会忘掉吧。不, 其实我们也很想忘掉。就彼此都把它忘掉吧。
- 基于上述的理由, 你们必须接受我们的挑战……开战吧!

规则如下, 一张 n 格长纸条, 格子内从左到右依次写着 $1, 2, \dots, n$ 共 n 个数字, 每次可以按数字之间的分界线折叠(常规意义下的纸片折叠, 与现实完全相同), 比如 $1, 2, 3, 4, 5$ 折一下变成 2 层, 上层 $3, 2, 1$, 下层 $4, 5$ 这样的。最后叠来叠去变成 n 层, 从上到下形成一个序列。

游戏程式每次会给 *Haruhi* 一个序列, 如果 *Haruhi* 能折出来就算她赢啦, 但是折不出来的话就要满足电研社以上的所有要求。

电研社可能会为了夺回电脑不择手段, 所以给出的序列并不一定能够折出来, 我们需要帮助 *Haruhi* 判断这个结果序列是否合法。虽然我还是很希望电研社能赢的, 但是为了世界的和平, 拜托啦!

Input

第一行一个正整数 $T(1 \leq T \leq 1000)$, 表示接下来有 T 组数据。

每组的第二行一个正整数 $n(1 \leq n \leq 80)$ 。

每组的第三行有 n 个正整数 $a_1 - a_n$ 表示结果序列, 数据保证这 n 个数是 $1 - n$ 的一个排列。

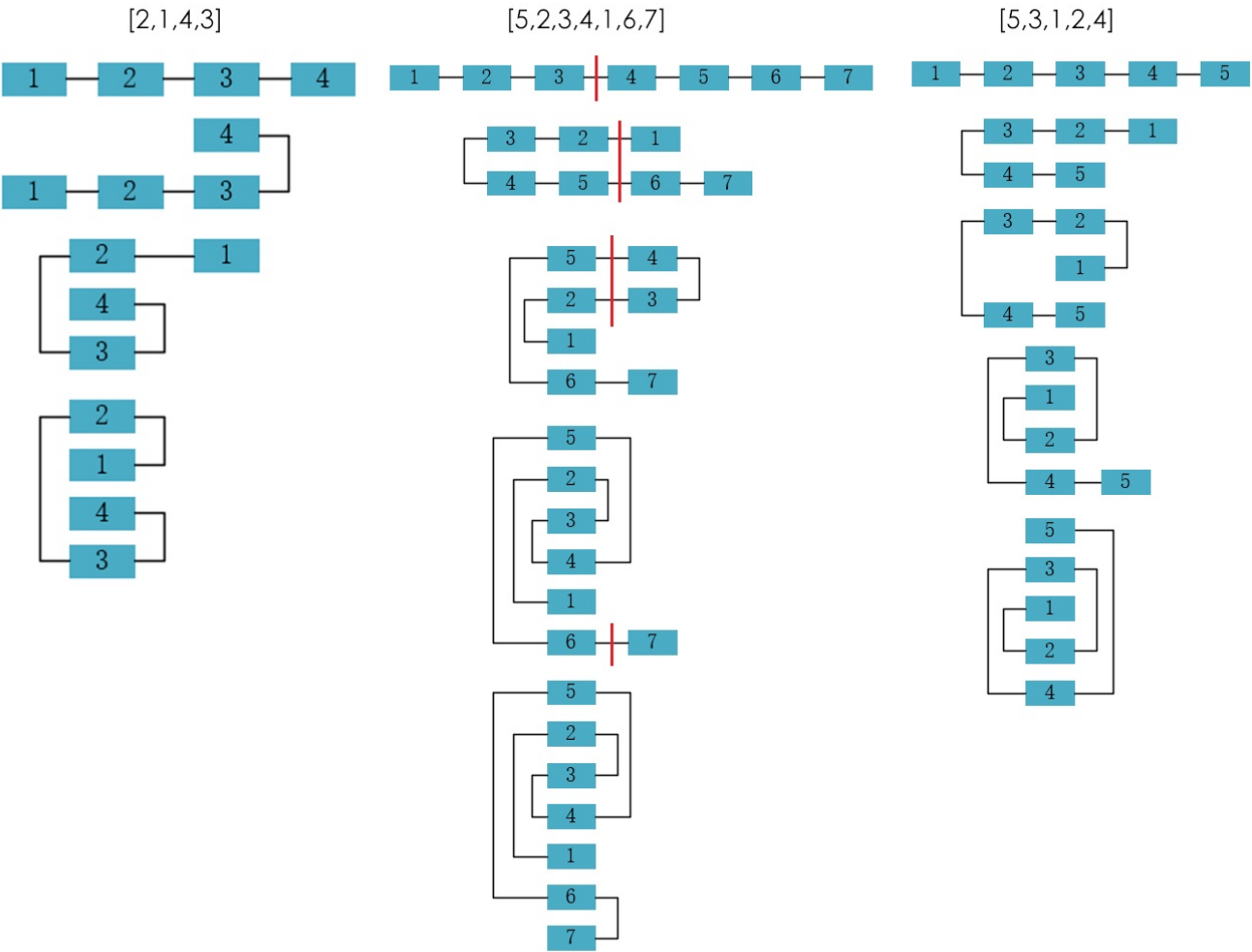
Output

对于每组数据输出一行 *yes* 或 *no* 表示是否能由原纸片折叠得到。

Sample

standard input	standard output
6	yes
5	yes
1 2 3 4 5	no
5	yes
5 3 1 2 4	yes
4	yes
1 3 2 4	
4	
2 1 4 3	
7	
5 2 3 4 1 6 7	
7	
5 2 3 4 1 7 6	

Note



Problem K. Haruhi with K-intimacy

Input file: standard input

Time limit: 2 seconds

Output file: standard output

Memory limit: 1024 megabytes

NagatoYuki，其真实身份为资讯统合思念体所创造出来的对有机生命体接触用人形联系装置，目的为观察 *Haruhi*。

“所谓的资讯统合思念体就是，在银河系，甚至是全宇宙如此广大的资讯大海里，存在着许多没有肉体的超高知性资讯生命体。他们最初是以资讯的型态诞生，然后各种资讯相互结合后会产生意识，最后再靠撷取其他资讯进化。没有实体，只能以资讯型态存在的他们，就算用最先进的光学检验方式，也完全无法观测出来。几乎和宇宙同时诞生的他们，随宇宙的膨胀而扩大，相对的资讯网也逐渐宽广、巨大化。对于最早在地球，不，应该说太阳系形成前的远古时代，便熟知全宇宙的他们来说，这颗位于银河系边境的星球根本没什么特别。因为存在着有机生命体的星球，除了这里以外还有许多。数也数不清。不过，随着这颗第三行星上进化成功的两足动物，萌发了名为知性的思索能力后，目前栖息在行星上的生命体使这颗被称为地球的酸性行星变得日渐重要。” *Nagato* 面无表情的解释道。

就算你这么说，我还是不懂啊！

唯一清楚的是，这个什么资讯统合思念体有着森严的等级制度。 i 与 j 的关系亲密度为 k 当且仅当 S_i 和 S_j 的最大公共前缀长恰好为 k ，其中 S_i, S_j 是 i 与 j 的名字，记为 $g(i, j) = k$ 。

现在定义 $f(x)$ 为在给定资讯统合思念体中，亲密度为 x 的有序对 $(i, j) (1 \leq i, j \leq n)$ 的数量。

Nagato 想要知道亲密度在 $[l, r]$ 里的有序对对数有多少对，即求 $\sum_{i=l}^r f(i)$ 。



Input

输入文件很大，请使用较快的输入方式

第一行两个正整数 $n, m (1 \leq n \leq 10^4, 1 \leq m \leq 10^5)$ ，表示资讯统合思念体的个数和询问个数。

接下来 n 行，每行一个字符串 $S_i (1 \leq |S_i| \leq 100)$ ，表示第 i 个资讯统合思念体的名字，仅包含大写英文字母。

接下来 m 行，每行两个正整数 $l_i, r_i (1 \leq l_i \leq r_i \leq 100)$ 。

Output

对于每个询问输出一行一个正整数。

Sample

standard input	standard output
3 3	9
AAA	9
AAA	5
AAB	
1 3	
2 100	
3 3	

Note

$$g(1, 1) = 3; g(1, 2) = 3; g(1, 3) = 2$$

$$g(2, 1) = 3; g(2, 2) = 3; g(2, 3) = 2$$

$$g(3, 1) = 2; g(3, 2) = 2; g(3, 3) = 3$$

$$\text{所以 } f(1) = 0; f(2) = 4; f(3) = 5$$

Problem L. Haruhi with Literature club

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

Doki Doki Literature Club

Haruhi 从 I 题的老板那里带回来许多扫地机器人，她准备用它们打扫旧校舍兼社团大楼里文艺部门前的走廊。

文艺部外的走廊可以看做 $1 * n$ 的平面网格图， k 台扫地机器人需要打扫这一行 n 格地板，每台机器人初始在第 a_i 格(无重叠)。

每台机器人每单位时间可以执行以下动作中的任意一条：

- 往左移动 1 个格子并打扫那个格子(n 只能到 $n - 1$)
- 往右移动 1 个格子并打扫那个格子(1 只能到 2)。
- 待在原地不动(原先所在的地砖可以视作已打扫)。

机器人互相之间独立不影响。

最终每台机器人都要回到初始位置，并且所有地砖都被扫到至少一次，Haruhi 想要知道最少需要花费多少时间。

Input

第一行两个正整数 $n, k(2 \leq n \leq 10^9, 1 \leq k \leq \min(2 * 10^5, n - 1))$ 。

第二行 $a_1 - a_k(1 \leq a_1 < a_2 < \dots < a_k \leq n)$ ，表示第 i 台扫地机器人的初始位置。

Output

一个整数表示最少花费时间。

Sample

standard input	standard output
5 3 1 3 5	2

Note

一种可行方案是：

第一台 $1 \rightarrow 2 \rightarrow 1$ 。

第二台可以 $3 \rightarrow 2 \rightarrow 3$ 也可以 $3 \rightarrow 4 \rightarrow 3$ ，或单纯待在原地。

第三台 $5 \rightarrow 4 \rightarrow 5$ 。

Problem M. Haruhi with Mogician

Input file: standard input

Time limit: 1 second

Output file: standard output

Memory limit: 256 megabytes

Haruhi 偶遇一位长者。

长者称他和质数很有缘分，就连他的生日都是质数，可以传授 *Haruhi* 一点人生的经验。

他告诉 *Haruhi*，区间 $[1, n-1]$ 中和 n 互素的数的个数可以揭示生命的奥秘，你的任务就是帮助 *Haruhi* 求出这个数。

签到成功 这是你的
签到奖励



Input

一行一个正整数 n ($2 \leq n \leq 10000$)。

Output

一个整数，表示区间 $[1, n-1]$ 中和 n 互素的数的个数。

Sample

standard input	standard output
6	2

Note

a 与 b 互素当且仅当 $\gcd(a, b) = 1$ ，即 a 与 b 的最大公约数是 1。