



任务 A：瘟疫时期的 AMPPZ

时间限制：12 秒，内存限制：1GB：12 秒，内存限制：1GB。

在瘟疫时期组织学术团队编程锦标赛是一项相当大的挑战。作为首席距离裁判，您的任务是确保参赛者之间保持适当的距离。来自同一所大学的参赛选手实际上就像家人一样，因此您主要关注的是来自不同大学的学生之间的距离。凭直觉，您希望每所大学的代表都聚成一个紧凑的小组，与

其他小组保持适当的距离。为了正式描述你的直觉，你引入了以下规则。让 A 表示属于同一所大学的两名学生之间的最大距离（欧氏距离，即平面上的标准距离），而 B 表示属于不同大学的两名学生之间的最小欧氏距离。这样就一定有 $A < B$

各位来宾热切地接受了这些建议，并在整个活动中遵循这些建议。遗憾的是不幸的是，有一个小问题：比赛结束后，您被要求证明社会距离原则确实得到了遵守。现在大家都已经回家了，你唯一要做的就是用一张集体照作为证明。问题是，你不知道哪些队员来自哪些大学。但既然你知道确实遵循了社会距离原则，也许你可以重新构建大学划分？

知道照片中所有学生的位置（描述为平面上的点⁽¹⁾）和大学数量，找到一个尊重您的原则的分部。**每所大学必须至少有一名学生。此外，您还可以假设解总是存在的。**

输入

输入的第一行包含数据集的数量 z ($1 \leq z \leq 100\,000$)。然后按以下形式依次给出数据集：

数据集的第一行包含两个整数 n, k ($2 \leq n \leq 2\,000\,000$, $2 \leq k \leq \min(n, 20)$)，分别表示学生数和大学数。

下面每行 n 都包含两个整数 x_i, y_i ($0 \leq x_i, y_i < 10^9$)，表示 i -th 学生的坐标。**没有两个学生站在同一点上。**

所有数据集中的学生总人数不会超过 10^7 人。

输出

对于每个数据集，输出 n 的数字 c_1, \dots, c_n ($1 \leq c_i \leq k$)，其中 c_i 是 i -th 学生所在大学的编号。学生与大学的分配应符合上述社会距离原则。如果有多种解法，可以写出其中任何一种。

¹集体照是用无人机从高空垂直拍摄的，因为从这个角度看，选手们最有优势。

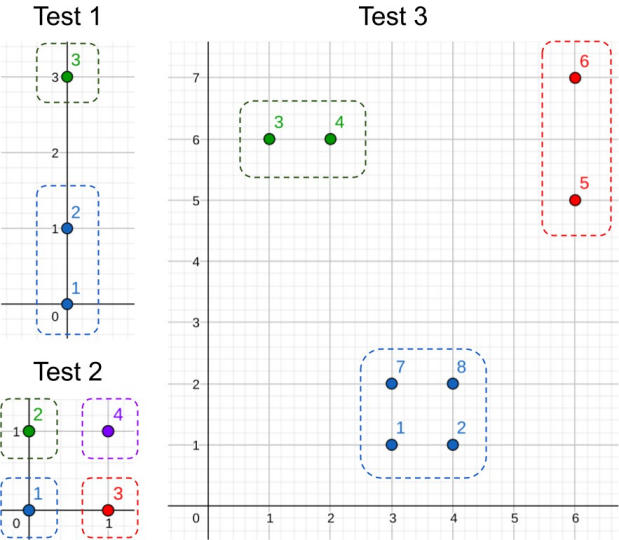


示例

输入数据:	正确答案可能是
3 3 2 0 0 0 1 0 3 4 4 0 0 0 1 1 0 1 1 8 3 3 1 4 1 1 6 2 6 6 5 6 7 3 2 4 2	1 1 2 4 1 3 2 2 2 1 1 3 3 2 2

注: 为便于阅读, 在示例测试输入中添加了空行。在运行解决方案的测试文件中不会出现这些空行。

下图为带标记答案的测试示例。





任务 B：外婆和饺子

时间限制：15 秒，内存限制：1GB：15 秒，内存限制：1GB。

祖母 Bytmila 决定举办一个宴会，在宴会上她将展示她最好的熟食产品--饺子。宴会上将摆放 n 个摊位，其中 i 个摊位上将摆放一个装有 p_i 个饺子的盘子，所有饺子的数量都不相同。这项任务似乎超出了一位老太太的能力范围，但 Bytmila 很快就完成了任务，她按照计划准备了 n 盘子，分别装有 p_1, p_2, \dots, p_n 的饺子。不幸的是，她在匆忙中把工位弄错了，盘子的摆放顺序也完全不同。

拜特米拉已经很累了，而且她害怕在这种情况下迷失方向。她不想移动盘子本身，但她可以把饺子从一个位置移动到另一个位置，使它们交换位置。更确切地说，她可以通过一次移动，选择分别装有 x 和 y 饺子的两个位置，然后在它们之间准确移动 $|x - y|$ ，这样第一个位置现在是 y ，第二个位置是 x 。每次这样的操作将花费她整整

$C + |x - y|$ 秒（ C 找勺子，每移动一个饺子用 1 秒）。

聚会就要开始了！奶奶不让你碰厨房里的任何东西，但有一件事你可以做

但有一件事你可以做：指定一连串的替换，以纠正这种情况，在最短的时间内，在每个工位做出正确数量的饺子。

输入

输入的第一行包含数据集的数量 z （ $1 \leq z \leq 1000$ ）。之后，数据集以如下形式依次给出：

数据集的第一行包含两个整数 n 和 C （ $1 \leq n \leq 200\,000$ ， $1 \leq C \leq 10^9$ ）。

该集合的下一个 n - 行包含职位描述。 i - 行包含两个整数

a_i 和 p_i （ $1 \leq a_i, p_i \leq 10^9$ ），分别表示 i -th 位置上当前和计划的饺子数量。

在每个集合中，数字 a_i 都是不同的。还知道集合 $\{a_1, a_2, \dots, a_n\}$ 和 $\{p_1, p_2, \dots, p_n\}$ 是相同的。

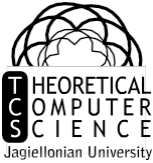
所有集合中 n 值的总和不超过 $1\,000\,000$ 。

输出

对于每个数据集，在第一行输出两个整数 S 和 K -- 分别是您的解决方案中的总时间和操作次数。

接下来的 K 行应描述解决方案的后续操作。在 k -th 行中写出两个数字 x_k 和 y_k ，表示 k -th 操作是在 x_k 和 y_k 之间转移饺子。完成所有操作后，每 i -th 位置上必须有 p_i 个饺子。

如果存在多个可能的交换序列，且总成本等于 S ，您可以列出其中任何一个。



示例

输入数据：	可能的正确答案是
1 4 2 2 4 3 2 1 1 4 3	6 2 2 1 4 1

说明

看台上有 2、3、1 和 4 个饺子。我们想把它分别变成 4、2、1 和 3。

在第一步棋中，我们交换一号和二号位置上的饺子。这次调换的代价是常数 $C = 2$ 和差值 $|3 - 2| = 1$ ，共计 3 秒。交换后，饺子的位置顺序为（3、2、1、4）。第二步，我们交换第一个和最后一个位置上的饺子。代价是 $2 + |4 - 3| = 3$ ，或再次耗时 3 秒。现在的顺序是（4、2、1、3），这是我们想要的顺序。替换的总成本为 6 秒，这是可能的最低成本。



任务 C. 蛋糕蛋糕

时间限制：15 秒15 秒，内存限制：1GB。

祖母拜特米拉 (Bytemila) (在任务 B 中已经为大家所熟知，她因出色的饺子而出名) 这次烤的是芝士蛋糕。她把芝士蛋糕切成 $2n$ 块长方形 (两排 n 块)，然后在每块蛋糕上涂上她选择的一种颜色的糖霜。她得意洋洋地看着自己的杰作，不禁惊呆了：拼凑起来的颜色看起来糟透了。

Bytemila 决定把这些颜色重新排列，让它们看起来更漂亮。但是，把面团一块一块地调换是不可能的：从芝士蛋糕中取出一块面团无疑会导致蛋糕边缘破碎。谁知道呢，但 Bytemila 的祖母绝不会让客人吃到切得不均匀的蛋糕！

幸运的是，Bytemila 有一把长方形的蛋糕铲，正好能放四块蛋糕 (分两排，每排两块)。使用它，她可以小心翼翼地将这四块蛋糕从面团中取出，然后转动抹刀，将它们从反面放回原处。这样的操作可以说是挑出某个 2×2 的正方形，然后将其旋转 180 度。

有了经验之后，你的祖母立刻找你帮忙，让你确定最小可能的移动次数，将颜色排列变成更漂亮的排列。当然，你做了任何人在你的位置上都会做的事情：你回答说你不会承担这项任务，因为要求的规格很模糊。然而，Bytemilla 并没有气馁：她在黑板上画了一个特别漂亮的颜色马赛克 (大小也是 $2 \times n$)，并要求你们确定最少要走多少步才能使芝士蛋糕完全变成这个样子。奶酪蛋糕的最小步数。

当然，也有可能是您的祖母 (在任务 B 中疲于搬运饺子) 犯了错误，黑板上画的颜色排列根本无法按照描述的方式实现。在这种情况下，你也必须尽快告诉她。

输入

输入的第一行包含数据集的数量 z ($1 \leq z \leq 10\,000$)。然后按以下形式依次给出数据集：

数据集的第一行包含整数 n ($2 \leq n \leq 500\,000$)。接下来的两行输入描述了两行芝士蛋糕的初始颜色。每一行都包含 n 范围为 $[1, 10^9]$ 的整数，中间用单个空格隔开。请注意，颜色可以多次使用，因此标识符可能会重复。

接下来的两行以相同的格式描述目标颜色布局。

所有 n 数据集的所有数值总和不超过 $2\,000\,000$ 。

输出

对于每个数据集，输出一个数字 - 实现所需颜色布局所需的最少操作数。如果无法实现所需配置，则输出 1。 -



示例

对于 输入数据:	正确答案是
2 4 1 2 3 2 4 3 1 3 3 2 1 1 2 3 4 3 2 1 2 3 4 3 4 1 2	3 -1

说明

在第一次测试中，芝士蛋糕的初始配置为

1 2 3 2
4 3 1 3

我们可以通过以下三个步骤来实现理想的配置：

1. 将最左边的正方形倒置、

3 4 3 2
2 1 1 3

2. 倒转最右边的正方形、

3 4 3 1
2 1 2 3

3. 中间正方形的倒置。

3 2 1 1
2 3 4 3

在第二个测试中，第一个配置不能以任何方式延续到第二个配置。



任务 D：两部分机制

时间限制：10 秒，内存限制：1GB：10 秒，内存限制：1GB。

Bytegate 机器是 Byteasar 的最新发明，它由两个部分组成，用兴奋的 Byteasar 的原话说，"这两个部分根本无法分开"！"他两岁的儿子拜特盖特马上就要让他知道自己错得有多离谱。

我们可以把拜特盖特的初始状态描述为一个大小为 $n \times m$ 的数组，其中充满了 A、B 和点。字母 A 表示机器的第一部分，字母 B 表示机器的第二部分，而点则表示机器的第三部分。

点则表示空格。下图是一个示例：

B	B	A	A	.
.	B	B	A	A
A	.	B	B	A
A	.	.	B	A
A	A	A	A	A

此外，机器的两个部分都是一致的，即对于任何两个标有 A 的字段，都有一定的路径连接这些字段，其中每个后续字段都与前一个字段共用一条边，并且路径上的每个字段都标有 A。

机器的 A 部分静止不动，而 Byteie 将 B 部分推向不同的方向。因此，他的游戏可以描述为一个由字母 N、S、E、W（分别表示北、南、东、西四个方向）组成的序列 q 。拜蒂每次都 "尽可能地" 推动机械装置的 B 部分，也就是说，如果再往前推，就意味着机械装置的两个部分要重叠在一起。可能出现的情况是，Byteie 可以无休止地移动 B 部分。

在这种情况下，我们可以说他成功地将两个部分分开了。但这并不意味着拜蒂在此时停止了对机械装置的抽动。尽管如此，我们还是要假设，一旦机器的两部分分离，它们就会一直保持这种状态，直到拜蒂的表演结束。

请帮助我们确定拜蒂是否在演奏过程中将字节门的两个部分分开。

输入

第一行输入包含数据集的数量 z ($1 \leq z \leq 1000$)。然后以下列形式依次给出数据集：

集合的第一行包含三个整数 n, m, q ($1 \leq n, m \leq 10, 1 \leq q \leq 100$)。接下来的 n 行输入描述了机器的初始状态。每一行都是长度为 m 的题词，由 A、B 和点组成。机器的两个部分都是非空且一致的。

集合的最后一行包含 q 字母序列，属于集合 $\{N, S, E, W\}$ ，其含义在任务正文中给出。

输出

对每组数据写出 "YES" 或 "NO"，表示 Byteie 是否成功分离了机器的两个部分。

**示例**

对于输入数据：	正确答案是
3 5 5 3 BBAA. .BBAA A.BBA A..BA AAAAA WNW 5 5 7 BBAA. .BBAA A.BBA A..BA AAAAA WNWSEN 6 5 3AAA. .A.A. .AB... .A.A. .AAA. SNE	否 是 .BBAA。

说明

当拜蒂玩完第一组和第三组的机器时，它们分别处于以下状态：

B	B
.	B	B	.	A	A	.
.	.	B	B	.	A	A
.	.	A	B	.	.	A
.	.	A	.	.	.	A
.	.	A	A	A	A	A

A	A	A
A	B	A
A	.	.
A	.	A
A	A	A

在第二组中，Bytie 在第四步棋中成功地将机制的两个部分分开。



任务 E：流行病

时间限制：9 秒，内存限制：1GB：9 秒，内存限制：1GB。

由于发现邻国感染了一种新的副细菌，拜特提亚关闭了边界。研究表明，这种菌株不仅具有高度传染性，而且不会引起拜特奥蒂亚人免疫系统的任何反应，因此一旦感染，一个人将终生受到感染，并传染给其他人（或至少在发明出有效的治疗方法之前）。

为了防止拜托特细菌的传播，拜托特政府出台了影响深远的限制措施，并启动了国家监控系统，对国内所有的社会接触进行监控。与此同时，政府还宣布，只有在确定除被隔离者外无人感染副杆菌后，才会解除限制。作为 Byteotia 的首席信息技术官，我们委托您对监控系统的数据进行分析，并确定何时可以解除限制。

拜托提亚领土上有 n 人，每个人最初都可能感染了该分支杆菌，也可能是健康的。边界关闭后， k ，每种情况都有以下几种形式：

- 您从监控系统中收到信息，称有一群人正在聚会。如果他们中有人受到感染，那么他们都会受到感染（而且会终生受到感染）。这种接触是感染的唯一可能途径（与最初的报道相反，接触受感染的表面不会感染拜托菌）。
- 某人接受了副结核杆菌检测，结果呈阴性。
- 某人接受了副结核杆菌检测，结果呈阳性。他（她）会立即被无限期隔离，从那时起将不会有任何社会接触（他（她）有可能在未来再次接受检测¹⁾）。
- 拜特卫生部长问您，是否已经可以解除限制，即根据目前收集到的所有信息，可以证明除了被隔离的人之外，没有人会受到感染。如果仍有感染者，您必须根据卫生部的指导原则（见出境部分）举例说明。

重要的是，您的程序必须是 *在线的*，也就是说，它必须在每位部长提出询问后立即给出答复，然后再加载其他询问。

¹⁾您可能会问，如果某人的检测结果呈阳性已成定局，为什么还要对其进行另一次检测？任务的作者已就此问题向拜特兰卫生部发出询问，但他们得到的信息是，由于案件性质复杂，答复时间已被延长三个月。



输入

输入数据的正确解释取决于 $shift$ 变量的当前值。在每个数据集开始时，它被设置为 0，其后续值将取决于程序给出的答案。这种输入描述的目的是迫使程序在加载每个查询后立即做出响应。

解码函数定义如下：

$decode(p) = ((p - 1 + shift) \bmod n) + 1$ 、

其中 p 是满足 $1 \leq p \leq n$ 的整数， \bmod 是余数除法运算。 n ， \bmod 是余数除法运算。

第一行输入包含数据集的数量 z ($1 \leq z \leq 1000$)。然后按以下形式依次给出数据集：

第一行包含两个整数 n 和 k ($1 \leq n \leq 500\,000$, $1 \leq k \leq 1\,000\,000$)，分别表示拜托提亚的人数和事件数。人的编号从 1 到 n 。

下面的 k 行描述连续的事件。它们的形式如下

- 字母 K 和整数 c ($2 \leq c \leq n$)，后面是 c 不同的整数 p_1, \dots, p_c ($1 \leq p_i \leq n$) - 一个社会联系人，后面是 c 的索引 $decode(p_1), \dots, (decode p_c)$ 。
- 字母 N 和整数 p ($1 \leq p \leq n$) - 一个索引为 $(decode p)$ 的人接受测试，结果为阴性。
- 字母 P 和整数 p ($1 \leq p \leq n$) -- 指数为 $(decode p)$ 的人检测结果呈阳性，将被隔离。您可以假定他/她从那时起不再参与任何社会交往（当然，将来对他/她进行的任何检测也会得出阳性结果）。
- 字母 Q 和整数 p ($1 \leq p \leq n$) -- 卫生部长的询问，其中的起始值（见输出部分）为 $(decode p)$ 。

所有集合中的数字 n 和 k 之和不超过 500 000，且 1 000 000。所有集合的所有查询中 c 的数字总和不超过 1 000 000。

输出

对于每个数据集，输出的行数与其中的查询次数（Q 事件）相同。

如果在 i -th 查询时，可以证明除被隔离者外没有人感染副杆菌，则在 i -th 行输出单词 YES²。查询结束后， $shift$ 变量的值将变为 0。

否则，在 i -th 行输出 NO 和一个人的标识符。如果

$decode(p)$ 是该查询的起始值，则必须输出字符串中第一个人的 ID ($(decode p)$, ($decode p + 1$, ..., n , 1, 2, ..., ($decode p - 1$)，它可能感染了副细菌，但未被检疫。

可能感染了副细菌但未被检疫。打印出来的数字将成为 $shift$ 变量的新值。

²可能发生的情况是，Byteotia 的所有居民都被检疫了。显然，在这种情况下，这句话将变为空句，您的程序也应输出 "是"。



备注

- 对于 K、N 和 P 类型的事件，*shift* 变量的值不会改变。
- 输入数据中给出的阳性和阴性检测结果总是描述了一种合理的情况，也就是说，对于每个数据集来说，至少有一组可能的初始感染者，而给出的数据并不包含矛盾之处。

示例

对于输入数据：	正确答案是
1 6 14 K 3 3 4 5 K 2 6 5 N 3 Q 3 P 1 K 2 6 2 P 6 Q 4 P 6 K 2 1 3 N 3 Q 4 N 2 Q 1	否 5 NO 1 是 是

- 解码后，上述示例如下：

```

1
6 14
K 3 3 4 5
K 2 6 5
N 3
Q 3
P 6
K 2 5 1
P 5
Q 3
P 1
K 2 2 4
N 4
Q 5
N 2

```





说明

- 在第一次查询之前，受试者 3、4 和 5 会面，然后受试者 5 和 6 会面，之后受试者 3 的测试结果为阴性。当收到第一次查询时，我们可以根据目前收集到的信息得出以下结论（从查询的起始值 3 开始）。人 3 此时一定是健康的（他们刚刚收到了阴性检测结果）。4 号人也必须是健康的（他或她在第一次相遇时不可能生病，因为他或她会感染 3 号人和 5 号人，这与 3 号人后来的阴性检测结果相矛盾；自那次相遇后，4 号人不可能被感染）。人 5 有可能被感染（在与人 3 和人 4 接触时，他一定是健康的，但后来遇到了人 6，我们无法从收集到的信息中推断出他的健康状况）。因此，查询的答案是 NOT 5，*移位*变量的值为 5。
- 然后，6 号人收到一个阳性检测结果，之后 1 号人和 5 号人相遇，接着 5 号人收到一个阳性检测结果。当收到第二个查询时，我们可以根据目前收集到的信息得出以下结论（从起始值 3 开始）。3 号和 4 号人仍然需要保持健康。5 号和 6 号处于隔离状态。第 1 人可能受到感染（此外，我们甚至可以推断出第 1 人目前一定受到感染，但任务中不需要解决这个问题）。因此，查询的答案是 NO 1，而 *shift* 变量的值为 1。
- 然后，1 号收到阳性检测结果，之后 2 号和 4 号相遇，然后 4 号收到阴性检测结果。当收到第三个查询时，我们可以根据目前收集到的信息得出以下结论（从起始值 5 开始）。5 号、6 号和 1 号人处于隔离状态。由于 4 号人的检测结果为阴性，因此 2 号人一定是健康的。3 号和 4 号一定是健康的。因此，查询的答案是 "是"，因为可以证明隔离区外的所有人（2、3 和 4）都是健康的。*移位*变量的值为 0。
- 第三次查询后，2 号人得到的检测结果为阴性。不难看出，这部分输入实际上是无关紧要的：在第一个 "是" 的回答之后，疫情已经被控制住了，必须重复 "是" 的回答，直到数据集结束。



任务 F：绘画

时间限制：3 秒，内存限制：1GB3 秒，内存限制：1GB。

人们普遍认为 Byteasar 是整个市最大的守财奴。有很多例子可以证明他的说法，其中最不重要的是他的房产连围墙都没有。不过，最近他在地窖里发现了 n 旧木板，于是他决定至少做一个栅栏。

Byteasar 把木板放在木桩上，使连续的木板长度为 a_1, \dots, a_n 。他拿起第一块木板，从上面切下一块 b 长的木板，钉在上面作为第一根栏杆，然后又从剩下的木板上切下一块 b 长的木板，钉在旁边。他继续这样做，直到手中剩下一块长度在 $[1, b]$ 之间的木板。拜塔萨想，这么好的木板不能浪费，尽管它看起来有点短。并把它钉在栅栏上，作为另一根栏杆。然后，他又从木板堆里拿了一块木板，接着又拿了一块，对每一块木板都重复了上述步骤。

完成工作后，拜特萨尔看着自己的作品，觉得钉上长短不一的栏杆可能不是最好的主意。他想，这看起来更像是随意堆砌的木板，而不是经过深思熟虑的结构。于是，他决定给栅栏刷上白色油漆，希望这样看起来至少专业一点。但是，过了一会儿，他意识到，如果我只把每一根栏杆都刷成白色，其余的都刷成棕色，那么我用的油漆（大约）就会是原来的两倍，而且栅栏还会给人一种连贯、周密的整体印象！

他是这样想的，也是这样做的，从第一根栏杆^①开始，他只给栅栏上的每一根栏杆刷了白漆。直到临睡前，拜塔萨尔才想到一个可怕的想法：如果他选择了不同长度的 b ，也许就能少用一些油漆？现在已经没有什么办法了，但意识到可能出错的拜特萨尔并不安心。所以他想知道，如果他选择了不同高度的栅栏，他需要用多少油漆。

帮助 Byteasar 解决这个问题，最终让他安然入睡（或不安宁，取决于您的计算结果）。

输入

输入的第一行包含数据集的数量 z ($1 \leq z \leq 5$)。之后，数据集按以下形式依次给出：
数据集的第一行包含整数 n ($1 \leq n \leq 1\,000\,000$)。第二行
第二行是 n 整数 a_i ($1 \leq a_i \leq 1\,000\,000$, $\sum_{i=1}^n a_i \leq 1\,000\,000$)，即
连续木板的长度。

输出

让 M 表示所选数据集中所有 a_i 值中的最大值。在该数据集的输出中，写出 M 行。 i -th 中应该有一个整数 f_i ：如果 Byteasar 决定建造高度为 $b = i$ 的栅栏，那么他必须把栏杆漆成白色的总长度。

^①如您所见，关于拜特萨尔吝啬的传言有些夸大其词。毕竟，他可以从第二根铁轨开始画画。



示例

输入数据:	正确答案是
1 4 10 7 2 8	14 13 15 13 15 16 21 23 24 12

说明

如果栅栏的高度是 $b = 4$ ，那么下列栏杆的高度是：4 4 2 4 3 2 4 4：Byteasar必须涂上长度分别为4、2、3和4的木板，因此第四行的答案是13。

对于栅栏的高度 $b = 5$ ，下列栏杆的高度是：5 5 2 2 5 3：Byteasar需要画长度分别为5、5、2和3的木板，因此第五行的答案是15。



任务 G: Gebalt 的诅咒

时间限制12 秒，内存限制：1GB。

和每年春天一样，巫师盖伯特踏上了他的旅途，他想利用自己的巫师技能，用普通人的便士塞满他的钱包。巫师的足迹从西向东延伸，长达 n ，每个马厩都有不同的东西在等着他，这些东西有三种性质：

- Bb_i ：凶猛野兽的巢穴，它无情地以农民为食。当巫师来到怪兽面前时，怪兽会抓住他，用爪子和獠牙伤害他，并 b_i 夺走他的生命。如果巫师的生命值为零，他就会死去；如果没有，他就会用刀砍向怪物，当场将其杀死。格巴吉特的生命值会发生如下变化

if $H < b_i$ then *death anticipated* else $H := H - b_i$ 。

- Kk_i ：村里的小酒馆，Gebajt（嗜酒如命）一定会去那里。如果他进入客栈时的活力低于 k_i ，他将不可避免地死于过度饮酒。否则，他会在黎明时带着诅咒离开客栈，寿命也会缩短到 k_i 。格巴吉特的寿命变化如下

if $H < k_i$ then *death anticipated* else $H := k_i$ 。

- Cc_i ：一个强大的女巫小屋，精通符咒和药水，可以治愈伤口和诅咒。如果盖伯特来到女巫那里时预期寿命低于 c_i ，她会将他的预期寿命恢复到 c_i 的水平。女巫的寿命变化如下

$H := \max(H, c_i)$ 。

巫师很自豪，因为他知道该走哪条路，这样既能体验到巫师的快乐，又能保住自己的生命。一天又一天过去了，在 i 这一天，有两件事发生了：

- 比如，当地商人买下了巫师的小屋，把它变成了一家客栈；或者，一只新的野兽从地里爬出来，用嘴里的火烧了客栈，并在那里建立了巢穴；
- 格巴赫特走到屋前，在他最喜欢的树下坐下，骄傲地说：如果从物
 l_i 如果他从目标出发，一路向东，他能走多远而不会丧命？盖巴特希望您能帮助他进行冥想，以使用密码秘术找到问题的答案。

请注意，巫师只是在想该怎么做，而不是真的在追踪，那么追踪中的变化将永久保留，但每个巫师独立于其他巫师的好奇心仍然存在，在每个生命周期中，Gebajt 在开始时的 H_0 单位是。

输入

输入的第一行包含数据集的数量 z （ $1 \leq z \leq 100\,000$ ）。之后，数据集以如下形式依次给出：

集合的第一行包含三个整数 n, q 和 H_0 （ $1 \leq n \leq 2\,000\,000, 1 \leq q \leq 4\,000\,000, 1 \leq H_0 \leq 10^{12}$ ）- 即足



迹的长度、天数和 Gebayt 的初始寿命。



该集合的下一行 n 描述了小路的初始状态； i -这一行包含与 i -这个对象的类型（B、K 或 C）相对应的字母，以及数字（ b_i, k_i 或 $c_i; 1 \leq i \leq n$ ）。 $b_i, k_i, c_i \leq 10^{12}$ ，其含义如上所述。

q 下一行描述每一天的情况。 i -th 行的开头字母是 Z（如果当天小路有变化），或者 D（如果有 Gebajt 假人）。

如果路径发生变化，该行其余部分包括：整数 x_i （ $1 \leq x_i \leq n$ ），表示被更改的对象；字母和数字，格式与初始状态描述相同，表示新的对象。对于 Gebajt 假人，该行包含一个整数 l_i （ $1 \leq l_i \leq n$ ），表示它开始徘徊的稳定位置。

轨迹总长度和总天数分别不超过 2 000 000 和 4 000 000。

输出

对于每个数据集，输出所有查询的答案。对于每个查询，输出一个整数，表示 Gebajt 可以到达（并活着离开）的最远对象 r_i （ $l_i \leq r_i \leq n$ ）的索引，如果他已经在 l_i 位置的冲突中死亡，则输出 1。- i -这一天的答案应考虑到前几天的所有变化。

示例

输入数据：	正确答案是
1	2
4 12 10	3
C 10	4
B 5	-1
K 5	3
B 6	4
Z 3 K 6	
Z 1 C 11	
D 2	
D 1	
Z 3 C 1	
D 3	
Z 3 B 20	
D 3	
Z 1 C 31	
D 1	
Z 4 K 6	
D 1	



说明

Gebayt 路线变化了六次，具体如下：

- [C 10, B 5, K 5, B 6] (开始)
- [C 10, B 5, K 6, B 6] (1 天)
- [C 11, B 5, K 6, B 6] (第二天)
- [C 11, B 5, C 1, B 6] (第 5 天)
- [C 11, B 5, B 20, B 6] (第 7 天)
- [C 31, B 5, B 20, B 6] (第 9 天)
- [C 31, B 5, B 20, K 6] (第 11 天)

Gebayt 在剩下的六天里感到自豪。

第三天，格巴吉特从第二个目标开始。打败野兽后，他只剩下 5 点生命值，比下一个目标（客栈 K6）的生命值少了 1 点。因此，他最远只能到达第二个目标。

第四天，格巴伊特从第一个目标开始；由于女巫的帮助，他多了 1 点生命值，在第三个目标（客栈）中存活了下来，但在最后一个目标（野兽）中失败了。

第六天，格巴伊特从第三个目标开始，现在是 C 1 级女巫。在生命值不变的情况下，他继续前进并击败了野兽（路径上的最后一个目标）。

第八天，格巴吉特从野兽 B 20 开始，他无法打败野兽 B 20，所以答案是-1。

第十天，格巴吉特从强大的 C 31 开始，用它打败了前两只野兽，但最后一只却失败了。

最后一天，Gebajt 可以骑完全程。



任务 H：密码

时间限制：5 秒，内存限制：1GB：5 秒，内存限制：1GB。

在成功实习后，Bytie 被聘为网络安全专家。为了给其他人树立一个好榜样，他决定给自己设置一个不同的电子邮件密码和一个不同的社交网站 *Facepalm* 密码。不幸的是，要记住两个不同的密码非常困难，而直接把密码写在纸上又违反了他自己设定的安全规则。于是，谨慎的拜蒂选择了一个秘密数字 $d > 0$ ，并把两个密码都写在纸上，用凯撒密码¹编码，偏移量为 d 。他满意地看了看纸片，大吃一惊：一旦编码，他的 *Facepalm* 密码就变成了 Mail 密码，反之亦然！"我做了什么……" - 他抓着头惊呼道。

也检查一下自己的网络安全！获得 Byte 第一个密码的数据后，重新创建第二个密码，或者确定无法明确创建。

输入

第一行输入包含数据集的数量 z ($1 \leq z \leq 20$)。然后以下列形式依次给出数据集：

每组密码由一个小写英文字母组成。一个密码最少有 1 个字符，最多有 200 000 个字符。

所有密码的长度总和不超过 1 000 000。

输出

针对每个给定密码，写出相应的第二个密码。如果第二个密码无法重现（解法不存在或有多个解法），则写出单词 NIE。

示例

输入	正确答案是
1 cnffjbeq	密码

¹凯撒密码是将每个字母替换成字母表中更靠前的 d 个字符，并对字母表进行循环包络。例如，对于 $d = 3$ ，字母 a 变为 d，字母 b 变为 e，。字母 w 变为 z，字母 x 变为 a，y 变为 b，z 变为 c。Byte 使用的是英文字母，不过凯撒更愿意称其为拉丁字母。



任务 I: 有趣的数字

时间限制10 秒，内存限制：1GB。

运算 \oplus 称为**比特排除和**或**比特 XOR**。它的工作原理如下：为了计算两个自然数的结果 $x \oplus y$ ，我们将这两个数都写入二进制系统，然后，当且仅当 x 和 y 的第 i 位数中正好有一位数等于1时，计算结果的第 i 位二进制数才等于1。换句话说，如果 x_i, y_i, z_i 分别表示 x, y 和 z 的第 i -位数，其中 $z = x \oplus y$ ，那么 $z_i = (x_i + y_i) \bmod 2$ 。我们从意义最小的数位开始编号。

给定一个正整数 k 。如果一个数列的每两个元素的比特 XOR 都不大于 k ，我们就称这个数列为有趣数列。在给出一个数列后，从中选择尽可能多的元素，这样所选的数列就构成了一个有趣数列。

输入

第一行输入包含数据集的数量 z （ $1 \leq z \leq 1000$ ）。然后以下列形式依次给出数据集：

集合的第一行包含两个整数 n, k （ $1 \leq n \leq 30\,000, 1 \leq k < 2^{20}$ ），表示字符串的长度和一个数字，指定两个元素的最大 XOR。

集合的第二行包含给定的字符串 s ，这些字符串由小于 2^{20} 的非负整数组成，中间用空格隔开。

所有数据集中的字符串长度总和不超过 200000。 k 之和不超过 3 200 000。

输出

为每个数据集输出一个整数，即从输入字符串中选取的元素的最大可能数目，以便它们组成一个有趣的序列。

示例

输入数据	正确答案是
1 7 11 3 12 9 10 16 3 4	4

说明

感兴趣的序列由元素 3、9、10 和 3 组成，因为每对元素的比特 XOR 都不超过 11。例如， $9 \oplus 10 = 1001_2 \oplus 1010_2 = 11_2 = 3 \leq 11$ 。不可能选择五个（或更多）具有此属性的元素：例如，序列（3、9、10、3、4）不合适，因为 $4 \oplus 9 = 100_2 \oplus 1001_2 = 1101_2 = 13 > 11$ 。



任务 J: 毒蛇

时间限制: 15 秒, 内存限制: 1GB。

毒蛇游戏棋盘是一个长方形, 有 n 行和 m 列, 分为 $n \cdot m$ 个单位的区域。每个区域可以是空的, 也可以被阻挡或包含毒蛇或无毒蛇的栖息地。玩家可以任意选择行数进行反转: 反转后, 该行中的每一个毒蛇栖息地都会变成非毒蛇栖息地, 反之亦然。同样, 玩家也可以将自己选择的列翻转。如果任何栖息地被翻转两次, 它就会恢复到原来的状态。完成所有这些操作后, 玩家必须从棋盘的左上方向右下方移动, 每次向右或向下移动一个区域。玩家的移动路径不能经过被封锁的区域或有毒蛇的区域。

游戏开发人员已经实现了 z 。不过, 仍有必要验证其中哪些棋盘是可以解决的。不幸的是, 这项任务交给了您。

输入

输入的第一行包含棋盘数量 z ($1 \leq z \leq 500$)。每块棋盘的描述如下:

第一行包含两个整数 n 和 m ($2 \leq m \leq 2000$)。

下面的 n 行中, 每一行都包含 m 个字符: .、#、O (大写字母 "o") 和 @ (猴子), 分别表示空域、封锁域、有无毒蛇类栖息的域和有毒蛇栖息的田地。您可以假设第一行的第一个字符和第 $m \cdot n$ -th line 的第 -ty 字符与 # 不同, 即左上角和右下角的字段均未屏蔽。

所有 n 值和 m 值的总和各不超过 15000。
每个。

输出

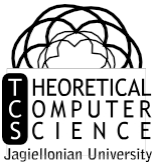
对于每组数据, 按以下格式输出解决方案。

在第一行输出 "是" 或 "否", 说明该棋盘是否可以解出。

如果您对给定集合的答案是 "是", 那么请按顺序写出以下三行:

- 由 n 个 T 或 N 字符组成的字符串, 其中 i -th 字符分别表示位于 i -th 行的栖息地颠倒或不颠倒;
- 由 m 个 T 或 N 字符组成的字符串, 其中 j -th 字符分别表示棋盘 j -th 列中颠倒或不颠倒的生境;
- 一个序列 $n + m - 2$ 个 P 或 D 符号, 其中从棋盘左上方区域开始的路径的连续步骤分别通向右边或向下。您所描述的路径必须通向棋盘的右下方区域, 并且只能使用空区域和有无毒蛇栖息的区域。

如果有多个正确答案, 您可以写出其中任何一个。



示例

输入数据:	可能的正确答案是
1 4 5 ..#.. @@0@@ ##@#0 ..@.@	是 NTNN NNTNT DPPDDPP

说明

将棋手指示的行列颠倒后，棋盘的状态如下：

..#..
0000@
##0#@
..0.0

棋手指示的路径只使用空地和（在所有反转之后）无毒蛇居住的空地。



任务 K: 猫和 Roomba

时间限制 12 秒，内存限制：1GB。

Bitusia 的猫 "船长" 最喜欢睡觉。不幸的是，自从 Bitus 决定购买机器人吸尘器 Roomba 后，它的睡眠质量明显下降。原来，猫队长害怕 Roomba，就像....。好吧，他只是非常害怕。

Bitus 的房子里有 n 个房间，连接着 $n-1$ 条双向走廊，这样就可以从任何房间到达任何其他房间。Bitus 注意到，如果 Roomba 进入船长所在的房间，猫就会醒来并逃到邻近的房间，然后马上又睡着了。受到惊吓的船长逃跑时完全是盲目的，因此，如果有多条走廊从房间里出来，它就有 *可能选择任何一条*（特别是，它可能会逃进 Roomba 刚刚离开的房间）。

在又一个漫漫长夜里，Bitty 打开 Roomba 应用程序，看到它在打扫卫生时陆续访问了 a_1 的房间... a_m 。这一连串的房间可以重复出现，每两个相邻的房间都有一条走廊相连。Bitus 还记得，在打开 Roomba 之前，猫咪正在 c 房间里睡觉。此外，还有 $a_1 \neq c$ ，因为谨慎的船长从不与 Roomba 睡在同一个房间！

现在 Bitty 想知道 Roomba 在打扫卫生时吵醒船长的次数的 *期望值* 是多少。帮他回答这个令人头疼的问题，这样他就能重新专注于工作了。

输入

第一行输入包含数据集的数量 z ($1 \leq z \leq 6\,000$)。然后按以下形式依次给出数据集：

集合的第一行包含两个整数 nc ($2 \leq n \leq 1\,000\,000$, $1 \leq c \leq n$)，分别表示比茨家的房间数和猫队长最初睡觉的房间数。

接下来的 $n-1$ 行描述的是走廊。每行包含两个整数 $u_i v_i$ ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$)，表示 u_i 和 v_i 这两个房间由一条走廊连接。可以假设从每个房间都可以到达其他房间。

集合的下一行包含 Roomba 访问过的房间数量 m ($1 \leq m \leq 5\,000\,000$)。

集合的最后一行包含 m 整数序列 a_i ($1 \leq a_i \leq n$) - Roomba 访问过的房间序列。每两个连续的房间都有一条走廊相连，也有一个 $a_1 \neq c$ 。

所有集合中 $n+m$ 的值总和不会超过 $12\,000\,000$ 。

输出

对于每组数据，写出一个实数 - 猫队长被 Roomba 唤醒次数的期望值。相对误差或绝对误差不超过 10^{-5} 即可认为答案正确。换句话说，如果你的算法答案是 a ，而正确答案是 b ，那么只要有 $|a-b| \leq 10^{-5}$ 就足够了。

$$\max(1, b)$$



例题

对 输入数据： 于	正确答案是
1 4 2 1 2 2 3 4 2 4 1 2 3 2	1.66666666666667



任务 L: 树懒

时间限制: 10 秒, 内存限制: 1GB: 10 秒, 内存限制: 1GB。

您即将开始一次丛林探险, 观察一种至今鲜有研究的树懒 *Choloepus manhattani*。树懒生活的整个丛林地区是世界上最奇特的地方之一: 那里的树木排列成一个完美的矩形 $n \times m$ 。在地图上, 它们被一对对自然数标记出来--树 (i, j) 生长在 i -th row 和 j -th column 的交叉点上。所有树懒 *Choloepus manhattani* 都生活在这一地区。

每只树懒都在其中一棵树上有自己的固定巢穴, 但有时也会离开巢穴到周围的树上寻找食物。在整个丛林中, 树懒只能从一棵树跳到另一棵树--在一次跳跃中, 树懒会跳到与前一棵树水平或垂直相邻的树上。为了不离开巢穴太远, 每只树懒只在离巢穴 k 跳的范围内觅食。换句话说, 如果树懒的巢穴在一棵树上 (x, y) , 那么树懒的觅食区域就是坐标为 (x', y') 的树的集合, 这些坐标满足 $|x-x'| + |y-y'| \leq k$ (and $1 \leq x' \leq n, 1 \leq y' \leq m$)。经过数百万年的进化, 所有树懒都有了不变的 k 。

你对这些神奇的报道有些怀疑, 但你无法与之前的研究者对质, 因为他有一天在丛林中莫名其妙地失踪了 (除其他外, 这让你重新思考树懒是否一定是食草动物.....)。他留下的只有一张尺寸为 $n \times m$ 的地图, 上面标注了树懒觅食的所有树木。然而, 它们的巢穴却没有标注在地图上。

检查该地图是否正确--确定是否有一组树懒的觅食区域与该地图完全吻合。

输入

输入的第一行包含数据集的数量 z ($1 \leq z \leq 4\,000$)。然后按以下形式依次给出数据集:

数据集的第一行包含整数 n, m, k ($1 \leq n, m, k \leq 1\,000$), 其含义在任务正文中给出。

以下 n - 行包含对地图的描述 - n - 每行中的字符。如果根据地图, 树懒在一棵树 (i, j) 上觅食, 那么在 i - 行的第 j - 个位置会有一个 `x`, 否则会有一个 `.` (点)。

$n+m+k$ 所有集合中的值之和不超过 100 000。

输出

对于每组数据, 写出一个 "是" 或 "否" 的标签, 表示对 "前一位研究者留下的地图是否能描述树懒的正确觅食区域" 这一问题的答案。



示例

对于输入数据:	正确答案是
2 3 3 1 .xx xxx xx. 3 4 1 ..xx x.xx x..x	是 无

说明

在第一个测试中，如果我们假设三只树懒栖息在田块（1，3）、（2，2）和（3，1）中，那么标记区域就是它们正确的觅食区域。

在第二个测试中，没有一组树懒栖息的田地会产生被标记的觅食区域。
标记的觅食区域。



任务 M：神奇三人组

时间限制：20 秒 20 秒，内存限制：1GB。

你正在参加考试，想知道如何确定序列的中位数。您费了九牛二虎之力才想起来，中位数是序列中不递减排序时占据中间位置的元素（如果序列的长度是偶数，中位数就是中间两个元素中较小的）。考试包括编写解决方案的伪代码，然后在教授给出的示例输入字符串上模拟操作。

尽管你的记忆漏洞百出，但你记得讲座中出现过类似的内容。不幸的是，你只记得讲座中模糊的片段（在 ByteStation 上度过的夜晚和算法教授单调的声音并没有帮助你集中注意力）。有一些神奇的算法……神奇的三？以某种方式分割字符串，递归调用，然后组合……？

根据你设法记住的片段，你想出了下面的算法：

函数 `magicThrees(string)`

如果字符串长度不大于 2，则返回字符串中最小的值

否则

`part_1, part_2, part_3 = divideNaThreeParts(string)`

`median_i = magicThrees(part_i)` for `i = 1, 2, 3` 返回字符串

`[median_1, median_2, median_3]` 的中值。

其中 `divideThreeParts` 将字符串分成三个长度尽可能相近的连贯片段。具体来说，连续片段的长度为 `[s, s, s]`、`[s+1, s, s]` 或 `[s+1, s+1, s]`，具体取决于字符串的长度。例如，字符串 `[8, 2, 6, 6, 3, 5, 7, 1]` 会分裂成 `[8, 2, 6]`、`[6, 3, 5]` 和 `[7, 1]`。

直到离开考场，你才意识到你的算法并不那么神奇，因为它并不总是有效。你满怀希望地想，也许至少它对任务中的序列是正确的……。不幸的是，你在这方面的记忆和算法本身一样模糊：虽然你几乎记得教授序列的所有元素，但对其中的一些值却不太确定。你只记得任务中给出的数据大小限制：序列中的所有数字必须在（封闭）区间 $[0, m-1]$ 。

请计算有多少种写法可以使您不知道的数字在使用 `magicTriple` 算法的情况下得出的序列能返回其真正的中位数（定义如上）。由于这个数字可能很大，所以只要给出除以 10^9 的余数+7 即可。

输入

第一行输入包含数据集的数量 z 。然后按以下形式依次给出数据集：

集合的第一行包含两个整数 $(nm \ n \otimes 1, 1 \ "m \ " \ 10^9)$ ，表示测试字符串的长度和元素值的约束条件

。

集合的第二行包含测试字符串，由 n 区间内的整数描述

$[-1, m-1]$ ，其中 -1 表示未知值的元素。



测试

如果我们用 q 表示未知值元素的数量，那么每个测试文件都属于以下三组之一：

- $1 \leq z \leq 100, 1 \leq q \leq 10, n \leq 3^4 = 81$
- $z \leq 15, 1 \leq q \leq 20, n \leq 3^5 = 243$
- $z \leq 3, q \leq 30, n \leq 3^8 = 6561$

输出

对于每组数据，输出一个整数 r ($0 \leq r < 10^9 + 7$) --任务正文中问题的答案。

示例

对于输入数据：	正确答案是
3	100
3 10	21
-1 -1 3	1979
4 50	
10 20 -1 40	
5 100	
-1 10 10 -1 20	

说明

在第一个测试中，无论序列中两个未知元素的值是多少，magicTrees 算法都会返回正确的中位数；因此答案是 $10^2 = 100$ 。

在第二个测试中，只有当未知值不大于 20 时，才会返回正确的中位数，这样就有 21 种可能。

在第三次测试中，如果两个未知值都小于 10，或两个未知值都大于 10，则 $100^2 - (10^2 + 89^{(2)}) = 1979$ 种可能性。