

小 B 的数字 (*numbers. opp/o/pas*)

Time limit : 1 Sec
Memory limit : 256 MB

Problem Statement

小 B 成为了一个数学家！他得到了一串数 a_1, a_2, \dots, a_n ，他想要得到另一串数 b_1, b_2, \dots, b_n ，其中 b_1, b_2, \dots, b_n 中每个数都是 **2 的正整数次幂**（即 2, 4, 8, ……），

使得对于 $\forall i \in \{1, 2, 3, \dots, n\}$ ，有 $\prod_{i=1}^n b_i \mid b_i^{a_i}$ 成立，即：令 $x = b_1 \times b_2 \times \dots \times b_n$ ，则有 $x \mid b_i^{a_i}$ 对于 $\forall i \in \{1, 2, 3, \dots, n\}$ 恒成立。他想知道这是否可能。可是他太小，不会算，请你帮帮他。

Input

从文件 *numbers.in* 读入数据。第一行一个正整数 T 表示数据组数。
对于每一组数据：
第一行一个正整数 n，表示两串数的长度。
第二行 n 个用空格隔开的正整数 a_1, a_2, \dots, a_n 。

Output

输出到文件 *numbers.out*。
对于每一组数据，输出一行 “YES” 或者 “NO”（不含双引号），表示是否可能找到满足题意的 b_1, b_2, \dots, b_n 。

Sample Input

```
3
2
3 2
3
3 3 3
2
```

1 10

Sample Output

YES

YES

NO

Explanation for the Sample

第一组数据，可以令 b 数组为 2, 2, 则 $x=4$ ，显然 $2^3, 2^2$ 均为 4 的倍数。

第二组数据，可以令 b 数组为 2, 2, 2, 易于验证其符合题意。

第三组数据，假定存在对应的 b 数组 α, β ，则 $\alpha\beta|\alpha$ ，所以 $\beta=1$ ，不是 2 的正整数次幂（注意 0 不是正整数），矛盾，故不存在。

Hint

对于 30%的数据，有 $n \leq 5, a_i \leq 3$ 。

对于 100%的数据，有 $n \leq 10^5, a_i \leq 10, T \leq 10$ 。

小 B 的矩阵 (*matrices.cpp/c/pas*)

Time limit : 1 Sec
Memory limit : 256 MB

Problem Statement

小 B 成为了一个扫雷大师！他遇到一种类似扫雷的新游戏：在一个 $n*m$ 的矩阵中 (n 为奇数)，每一格都有可能雷。不同于普通的扫雷游戏的是，每一格上都会显示一个数字，表示的是这个格子周围八个格子以及这个格子本身的一共九个格子中，一共有多少个雷，那么显然这个数字在 0-9 之间。小 B 毫无头绪，决定开始猜测。他发现正中间一行比较好猜，于是他会随机点击正中间一行的某一个格子。他想知道，对于给定的数字矩阵可能产生的每一种布雷方式中，使得他没踩中雷的概率最小那种布雷方式，没踩中雷的概率是多少。可是他太小，不会算，请你帮帮他。

Input

从文件 *matrices.in* 读入数据。每一个测试点只有一组数据。
第一行两个空格隔开的正整数 n 和 m ，表示矩阵的长和宽。
接下来 n 行，每行 m 个整数（之间没有空格隔开），描述了 Problem Statement 中讲述的数字矩阵。

Output

输出到文件 *matrices.out*。
共输出一行，表示所有布雷方式中，没踩中雷的概率最小是多少。由于小 B 不喜欢可能会产生误差的小数，所以请你输出为最简分数 p/q 的形式，特别地，若答案为 0，则输出 “0/1”（不含双引号）。

Sample Input

```
3 4
1211
2332
2221
```

Sample Output

3/4

Explanation for the Sample

可以证明，对于样例中的数字矩阵，正中间一行至多仅有 1 个雷，下面给出其中一种布雷方式：

```
..*.  
*...  
.*.*
```

其中. 表示空格，*表示雷。

Hints

对于 20%的数据， $n = 3, 3 \leq m \leq 7$ 。

对于 60%的数据， $3 \leq n, m \leq 50$ 。

对于 100%的数据， $3 \leq n, m \leq 10^3$ ，保证 n 为奇数，且输入的数字矩阵正确，即存在至少一种对应的布雷方式。

小 B 的咒语 (*strings.cpp/c/pas*)

Time limit : 1 Sec
Memory limit : 256 MB

Problem Statement

小 B 成为了一个魔法师！他得到了一串由正整数组成的咒语 a 。经过他的研究，某一个咒语的威力可以表示为，每个在此咒语中出现过的数字，在这个咒语中最后一次出现与第一次出现的位置之差。小 B 做出了若干次操作，每次操作都有可能是计算咒语的某个子串的威力。同时，小 B 也在不断地实验中，所以操作也有可能是把咒语中某个位置上的数字换成另一个。可是他太小，不会算，请你帮帮他。

Input

从文件 *strings.in* 读入数据。每一个测试点只有一组数据。
第一行有两个空格隔开的正整数 n 和 q ，表示咒语 a 的长度和操作次数。
第二行有 n 个空格隔开的正整数，用于描述初始的咒语 a 。

接下来 q 行，每行是用空格隔开的字符串和两个正整数 $type_i, x_i, y_i$ 。

若 $type_i$ 为 “Modify”（不含双引号），那么表示将咒语第 x_i 个位置的数改为 y_i 。

若 $type_i$ 为 “Query”（不含双引号），那么表示询问咒语第 x_i 个到第 y_i 个位置组成的子串的威力。

Output

输出到文件 *strings.out*。

对于操作中每次 Query 操作，输出一个正整数表示询问的子串的威力然后换行（参考样例）。

Sample Input

```
7 6
1 2 3 1 3 2 1
Query 3 7
Query 1 3
Modify 7 2
```

Modify 3 2

Query 1 6

Query 5 7

Sample Output

5

0

7

1

Explanation to the Sample

样例中的每次询问解释如下：

Query 3 7: $(7-4)+(6-6)+(5-3)=5$

Query 1 3: $(1-1)+(2-2)+(3-3)=0$

Modify 7 2: 数组变为 1, 2, 3, 1, 3, 2, 2

Modify 3 2: 数组变为 1, 2, 2, 1, 3, 2, 2

Query 1 6: $(4-1)+(6-2)+(5-5)=7$

Query 5 7: $(7-6)+(5-5)=1$

Hint

对于 20% 的数据, $n, q \leq 10^3$ 。

对于 50% 的数据, $n, q \leq 5 \times 10^4$ 。

对于另外 30% 的数据, $n, q \leq 10^5$, 保证只有 Query 操作。

对于 100% 的数据, $n, q \leq 10^5$,

$a_i \leq 10^5, type_i \in \{ "Query", "Modify" \}$,

且对于 Query 操作, $x_i \leq y_i \leq n$, 对于 Modify 操作, $x_i \leq n, y_i \leq 10^5$ 。