

Problem A. 今天不是周四

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

今天是周六，但前天是周四。一说到周四，Redcrown最怕听到的就是下面这句话：

FuckingThursdayVme50

出于极大的好心，每当有同学和他说这句话，他都会花 50 元请对方恰一顿KFC。看着越来越消瘦的钱包，他发誓，明年起不会再请了。



然而，今年还有一个月，所以请你帮他考虑一下，假如在今年里未来的某一天有 x 位同学给Redcrown塞了那句话，他那天会给出去多少钱。

Input

输入只有一行，有一个字符串 str 和一个非负整数 x ，字符串 str 形如 $YYYYMMDD$ ，表示今年某一天的年份、月份和日期，以及说这句话的同学人数。

输入保证题目给出的日期不早于今天 (20221126)，但早于2023年元旦，即 $YYYY = 2022$ ，若 $MM = 11$ 则 $26 \leq DD \leq 30$ ，否则 $MM = 12$ 且 $1 \leq DD \leq 31$ ，具体请见样例。另外还保证 $0 \leq x \leq 50$ 。

Output

一行一个整数，表示Redcrown会给出去多少钱。

Examples

standard input	standard output
20221126 1	0
20221201 2	100

Problem B. 趣味赛分组

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

第 13 届趣味赛正在进行，但由于选手们对算法竞赛的了解与掌握程度有差异，因此小 P 想要通过分组使得更多的同学能够体会到趣味赛的乐趣。

因此，小 P 想到了对趣味赛的选手进行分组，分组的依据是每个人在 OIwiki 上有个评分 a_i 且 a_i 一定为正整数。

这次比赛一共报了 n 位选手，每位选手的评分为 a_i ，现在小 P 想要把这 n 位选手分为 2 组，（每组人数不一定相同，但每组至少得有一人），分别称为"初学组"和"进阶组"。

小 P 希望初学组中最大的能力值与进阶组中最小的能力值之差的绝对值最小。

小 P 想知道这个差值最小是多少？

Input

第一行一个整数 T ，表示数据组数。

对于每组数据，第一行一个整数 n ，第二行 n 个整数 a_1, \dots, a_n 。

对于所有数据， $2 \leq n \leq 10^3, 1 \leq T \leq 5, 1 \leq a_i \leq 10^9$ 。

Output

对于每一组数据，输出一行一个整数，表示最小的差值。

Example

standard input	standard output
2	0
6	1
3 1 4 2 2 3	
5	
3 1 2 6 4	

Note

（分法不一定唯一）第一组样例，初学组为 $\{1, 2\}$ ，进阶组为 $\{2, 3, 3, 4\}$ ，差值为 $|2 - 2| = 0$ 。第二组样例，初学组为 $\{1, 2, 4\}$ ，进阶组为 $\{3, 6\}$ ，差值为 $|4 - 3| = 1$ 。

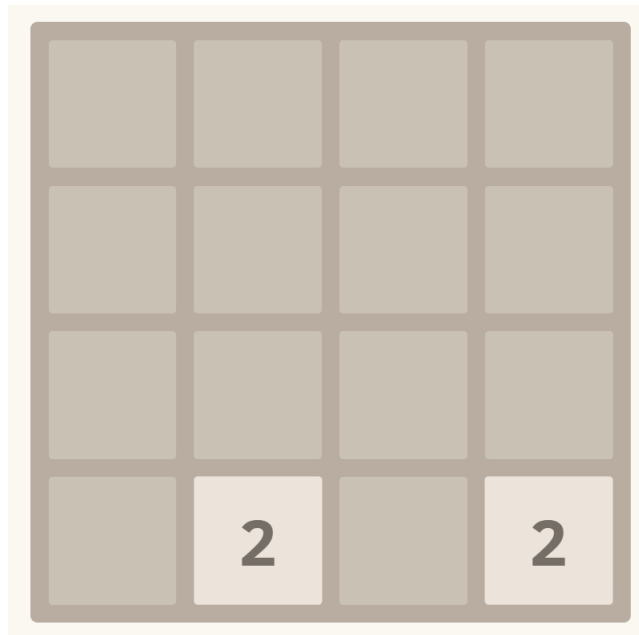
Problem C. New-2048

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

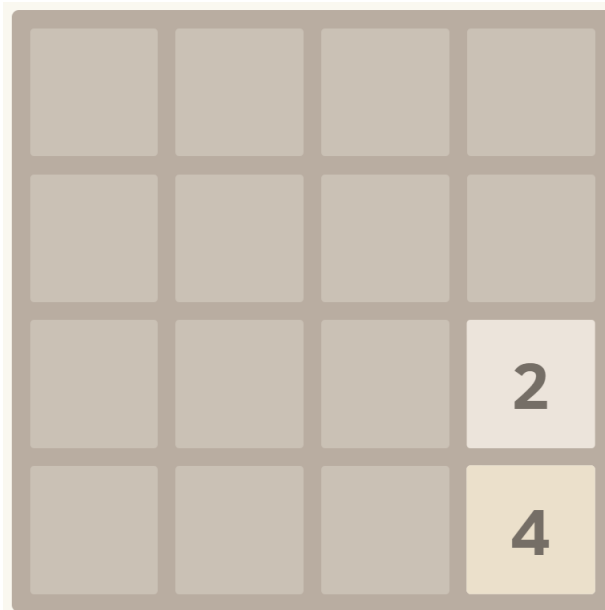
橙汁桃最近喜欢上一款经典游戏 2048，就是在一个 $n \times n$ 的格子图中上下左右移动合成数字，目标是合成出 2048，每次移动所有数字方块都会向移动方向滑去，直到被其他方块阻挡或者到达边界，如果两个相同的数字碰撞则会合成新的，新值的为原来两个方块的和，每当有方块合成或者移动，结束后就会在空的位置随机出现一个新的方块。

可是笨笨的橙汁桃怎么都合不出来，于是他去请教大佬游戏攻略。大佬告诉他，只需主要移动两个方向即可...，心急的橙汁桃并没有听完，听了一半就马上跑回去自己试试去了，他决定选择两个移动方向，然后一直轮流操作这两个方向直到无论多少次操作后都无法再有新的变化出现，可是出现的数字方块以及位置是随机的，他始终还是没法合成 2048。

于是他决定使用游戏修改器，可以修改下一次出现的数字方块上数字大小和位置，但似乎还是不能合成 2048，于是他转而想获得最高分数，可惜笨笨的他不会计算，所以请你帮他算算在他的策略下，最多能得多少分。



初始状态



往右操作一次

所得分数是最后状态 $n \times n$ 的格子中所有数字值之和，橙汁桃操作只会有两个不同方向交替执行，如 LR，则操作序列为 LRLRLRLR...，每次出现数字和位置都可以用修改器修改（包括初始状态时的一个方块）。

Input

第一行一个 T ，表示 T 组数据

每组数据格式如下：第一行 n, m ，表示2048游戏格子大小（ $n \times n$ 个）， m 表示出现的数字范围在 $[1, m]$ 之间的整数，接下来一行两个字母表示操作，字母均为 **L,R,U,D** 中，分别表示向左，右，上，下移动。

Output

对于每组数据，输出一行一个数字，表示能获得的最高分数。

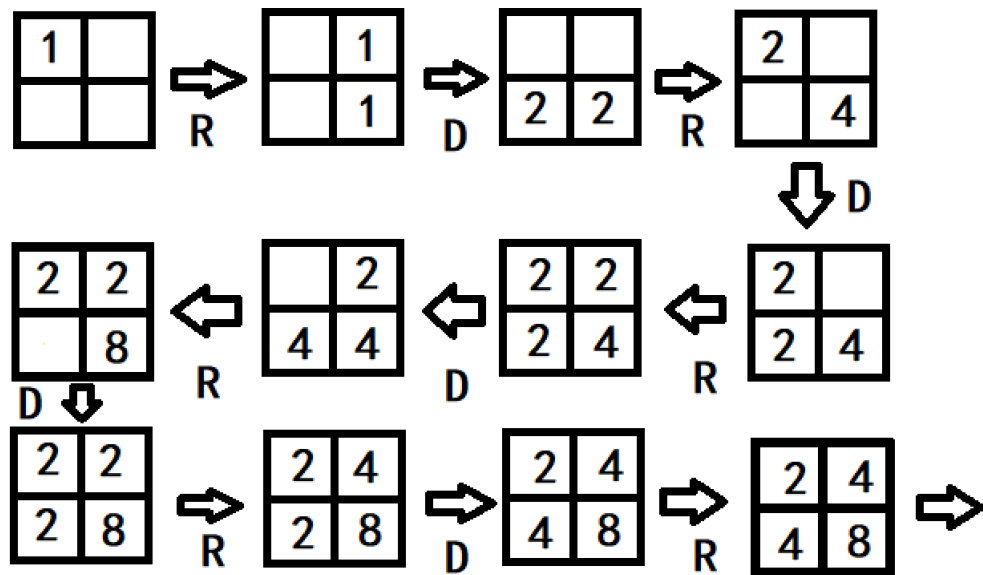
Example

standard input	standard output
4	18
2 2	2
RD	24
1 2	2048
LR	
2 4	
DU	
1 2048	
LU	

Note

$1 \leq T \leq 100, 1 \leq n \leq 15, 1 \leq m \leq 2048$

样例第一组的一种可行的方案：



可行方案

$$4 + 8 + 2 + 4 = 18$$

答案为 18。

注意：答案可能超出 `int` 范围。

Problem D. 羊了个羊

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

深呼吸，现在让我用十秒钟告诉你题意：现在Redcrown在玩羊了个羊，还剩4种12张卡，槽全空，没有道具，请问他能赢吗？



如果你还没理解这道题，别急，请深吸亿口气，放首《普通的disco》，让我慢慢告诉你羊了个羊怎么玩。

你现在手上有一个口袋，里面可以装7张卡。当你的口袋中存在三张一样的卡时，这三张卡就会被清除。为了简化，每张卡都是一个 1×1 的方形。卡牌之间可以相互覆盖。对于任何一张卡A，如果它不被任何一张卡覆盖，则称A为活卡，否则称为死卡。当一张死卡上方的所有卡被拿了之后，他会变成活卡。

游戏场可以看成是一张大坐标纸，为了方便，这里用12个数表示12个不同的可能有数的方格。场上有12张卡，一共四种卡，每种有三张，他们被分成了若干个互不重叠的堆。每张卡都和方格对齐，这些卡将被按顺序叠放在游戏场上。你可以做以下操作直到场上不剩卡：

- 任选一张活卡，将其从游戏场上清除，并将其放入你的口袋中。

若你的口袋中有7张卡且都不能被清除，你就输了。若游戏场上不剩任何一张卡，你就赢了。

众所周知，这个游戏很容易无解。现在Redcrown好不容易玩到只剩12张卡，面对着这个游戏场，如果他足够聪明，请问他有机会赢吗？

Input

输入有12行，第 i 行有一个整数 x 和一个大写字母 ch ，表示第 i 张卡所在的堆号和卡的种类，第 $i+1$ 张卡放在第 i 张卡之后。输入保证 $1 \leq x \leq 12$ ， $ch \in \{A, B, C, D\}$ ，且每种字母出现恰好三次。

Output

若Redcrown能赢，输出YES，否则输出NO，区分大小写。

Examples

standard input	standard output
1 A 1 B 1 C 1 D 1 A 1 B 1 C 1 D 1 A 1 B 1 C 1 D	NO
1 A 1 A 2 A 2 B 3 B 3 B 4 C 4 C 5 C 5 D 6 D 6 D	YES

Note

对于样例一，由于卡和卡之间的覆盖，这里你只能从第12张卡开始拿，然后拿第11张，第10张。。。最后你的口袋里会有七张卡，每一张卡都不能消除，故失败。

请注意样例一只有一叠卡，最上面的可以拿的是D。

对于样例二，要注意，和真正的羊了个羊不同，这个游戏的卡只有"重叠"和"相离"两种关系，例如第10张卡和第11张卡相离。

Problem E. 现在是! *Ain* 的自习时间

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

众所周知，泥电对大一新生提出了上早自习的要求。*Ain*，作为新生的一员，很是理解感激学校为了帮助大一新生尽快适应大学生活的良苦用心。但是在 *Ain* 看来，每天都去同一间自习教室显得有些无趣。因此，*Ain* 和他的朋友们决定采取一种不同的方式上早自习。

具体的，有 n 间早自习教室，编号从 1 到 n 。有 n 个大一新生要上早自习，编号也从 1 到 n ，其中有部分新生是 *Ain* 和他的朋友们。按学校规定，编号为 i 的新生应该在第 i 间教室上早自习。上早自习时，每个大一新生按编号从小到大的顺序进入教室。

- 对于 *Ain* 和他的朋友们，他们将会等概率的选择一间无人进入的教室上早自习。
- 对于其他新生，如果他对应的教室无人进入，那么就在这间教室上自习；如果对应的教室已经被占用，那么他会等概率的选择一间无人进入的教室上早自习。

现在 *Ain* 想知道，在上述策略下，编号为 n 的新生在 n 号教室上早自习的概率是多少？

Input

第一行两个整数 n ($1 \leq n \leq 10^5$)，代表新生总人数。

第二行一个长度为 n 的 01 串，表示编号为 i 的新生身份。如果第 i 位为 1，表示编号为 i 的新生是 *Ain* 和他的朋友们；若为 0，则为其他新生。

Output

输出一行共两个互质的整数，分别表示所求概率的分子和分母。

若概率为 0，输出 0 0。

若概率为 1，输出 1 1。

Example

standard input	standard output
3 101	1 2

Note

对于样例，共有三个新生，其中有两个是 *Ain* 和他的朋友。

如果第 1 个新生选择 1 号教室，则 3 号新生在 3 号教室，概率为 $\frac{1}{3}$ 。

如果第 1 个新生选择 2 号教室，则 2 号新生的教室被占用。此时如果 2 号新生选择 1 号教室，则 3 号新生在 3 号教室，概率为 $\frac{1}{3} * \frac{1}{2} = \frac{1}{6}$ 。如果 2 号新生选择 3 号教室，则 3 号新生无法在 3 号教室上早自习。

如果第 1 个新生选择 3 号教室，则 3 号新生无法在 3 号教室。

综上，答案为 $\frac{1}{3} + \frac{1}{6} = \frac{1}{2}$ 。

Problem F. 爱情游戏

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

安娜苏和徐伦在玩一个游戏。

给出一个长度为 n 的数组 x ，其中第 i 个元素为 x_i 。

给出一个 n 阶方阵 A ，满足

$$a_{ij} = \begin{cases} 1 + x_i & , i = j \\ x_i & , i \neq j \end{cases}$$

其中 a_{ij} 表示 A 中第 i 行第 j 列的元素。

游戏开始时，令矩阵 $B = A$ ，记矩阵 B 的阶数为 k 。

接下来两人轮流操作，在每次操作中

- 若 $k > 1$ ，玩家选择一个 $i (1 \leq i \leq k)$ ，得到 $|B| - |M_{ii}|$ 分，然后令 $B = M_{ii}$ 。其中 M_{ii} 表示 B 去掉第 i 行第 i 列得到的 $k - 1$ 阶子矩阵。
- 若 $k = 1$ ，玩家得到 $|B| - 1$ 分，游戏结束。

最终获得分数高的玩家获胜。注意：两位玩家分数相同时，双方都不视为获胜。

出于绅士风度，安娜苏决定让徐伦先手。现在他们希望你来判断，当两人都采取最优策略时，徐伦能否获胜？

Input

第一行输入一个正整数 $n (1 \leq n \leq 10^5)$ ， n 是偶数。

第二行输入 n 个正整数，第 i 个数为 $x_i (1 \leq x_i \leq 10^5)$ 。

Output

一行一个字符串 **YES** 或 **NO**，其中 **YES** 表示徐伦能获胜，**NO** 表示徐伦不能获胜。

Examples

standard input	standard output
6 1 1 4 5 1 4	YES
2 2 2	NO

Note

当一个矩阵 A 的行数 m 和列数 n 相等时，称 A 为 n 阶方阵，其阶数为 n 。

$|A|$ 表示矩阵 A 的行列式。

【样例2解释】

游戏开始时， $B = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$ 。

第一次操作徐伦选择 a_{11} 或 a_{22} ，结果是一样的：获得 2 分，并令 $B = [3]$ 。

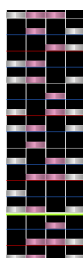
第二次操作安娜苏获得 2 分，游戏结束。

两人分数相同，徐伦不能获胜。

Problem G. 叠

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 256 megabytes

在传统下落式音乐游戏中，音符(note)在特定时刻出现在特定位置，玩家需要按正确的按键以获得分数。轨道是对应着相同按键的音符的轨迹。叠(Jack)，又称叠键，是一种在同一轨道连续出现多个音符的键型。



一个叠的例子

由于某些原因，玩家会形成错误的肌肉记忆，wwawwaww也一样。一条轨道的信息可以用长度为 n 的序列 $a = \{a_1, a_2, \dots, a_n\}$ 表示， a_i 表示第 i 个时刻是否存在音符。 $a_i = 0$ 表示第 i 个时刻无音符， $a_i = 1$ 表示第 i 个时刻有一个音符。对于一条特定的轨道，他记住了叠的压缩序列 $len = \{len_1, len_2, \dots, len_m\}$ ，过程如下：

1. 选出 m 个区间 $[l_1, r_1], [l_2, r_2], \dots, [l_m, r_m]$ ，满足：
 - $1 \leq l_1 \leq r_1, l_2 \leq r_2, \dots, l_m \leq r_m \leq n$;
 - 对于在 $[1, m)$ 的每个整数 j ，有 $r_j + 1 < l_{j+1}$;
 - 若下标 i 属于这些区间的并集，则 $a_i = 1$ ，否则 $a_i = 0$ 。

容易证明在上述条件下 m 和这些区间是唯一确定的。

2. 对于 $[1, m]$ 中的每个整数 j ，令 $len_j = r_j - l_j + 1$ 。

显然，可能存在多种不同的序列 a 能得到相同的序列 len 。wwawwaww现在只知道序列 len 和原始序列 a 的长度 n 。但是在得到 len 后又忘记了 a 中每个元素具体的值。他想求对 $[1, n]$ 中的每个整数 i ，能否确认 a_i 的值。为了尽快改掉这一不好的习惯和获得时间多练，他请您来解决这一问题。

Input

第一行包含两个整数 n, m ，分别表示序列 a 的长度和序列 len 的长度。第二行包含 m 个整数 $len_1, len_2, \dots, len_m$ ，第 i 个数表示 len_i 。
 $1 \leq m \leq n \leq 2 \times 10^5, len_1, \dots, len_m \geq 1, m - 1 + \sum_{j=1}^m len_j \leq n$.

Output

输出一个长为 n 的字符串 s ，若能确定 $a_i = 1$ ，则 $s_i = '1'$ ，若能确定 $a_i = 0$ ，则 $s_i = '0'$ ，否则 $s_i = '?'$ （下标从1开始）。

Examples

standard input	standard output
7 2 2 3	?1??11?
4 2 2 1	1101
6 3 1 1 1	??????

Note

对于第一个样例，所有可能的 a 为 $\{1, 1, 0, 1, 1, 1, 0\}, \{1, 1, 0, 0, 1, 1, 1\}, \{0, 1, 1, 0, 1, 1, 1\}$ 。 a_2, a_5, a_6 必定为 1，其余位置都可能有两种取值。