要文件读写，要建子文件夹，开O2,c++11，评测机很菜

写题面的是xpp，题面有锅来喷xpp

氨基酸序列

文件名：amino

在 S 星球，生物体需要的氨基酸只有两种：得氨酸和便氨酸。

因此，S 星球的蛋白质仅由这两种氨基酸构成肽链构成。我们用 0 和 1 分别表示得氨酸和便氨酸，那么生物体的蛋白质可以表示成一段由 0 和 1 构成的氨基酸序列。

Θ 国珂学院在研究完该冠状病毒的 RNA 序列后，转而开始研究~~更简单的~~氨基酸序列。

研究表明，该冠状病毒的氨基酸序列的长度为 L。也就是说，如果以氨基酸序列为标准，则一共有 2^L 种不同的冠状病毒。

8224 年 4 月 1 日，作为 S 星球疫情的一个转折点，她们已经对 SARS-CoV-233 病毒的氨基酸序列有了初步的了解，并打算在培养皿中进一步观察。

具体地，将在接下来的 q 天，依次发生如下事件，其中每天发生下面两种事件之一：

1. 培养皿中，氨基酸序列为 seq 的病毒的数量增加了 val，其中 $val > 0$ 表示增加了 val 个个体，$val < 0$ 表示减少了 $-val$ 个个体，$val = 0$ 表示没有发生变化。
2. 研究表明，给定一个只包含 $0/1/*$ 的序列 $mask$ (称为**致死掩码**)，如果氨基酸序列 seq 满足，对于 $mask$ 中每个非 $*$ 的位置，seq 中对应的氨基酸和 $mask$ 相同，则称该氨基酸序列是致死的。珂学家想知道现在的培养皿中有**多少个病毒**的氨基酸序列是致死的。

由于生物的进化，不同时间下致死掩码 $mask$ 是不相同的，你需要对这些不同的 $mask$ 分别作出回答。

输入格式

第一行包含两个正整数 L, q，分别表示氨基酸序列的长度和事件的个数。

接下来 q 行，每行描述一个事件，格式如下：

1. `I seq val` 表示氨基酸序列为 seq 的病毒数量增加了 val，其中 seq 为长度为 L 的 $0/1$ 串。
2. `Q mask` 表示对于致死掩码 $mask$，询问培养皿中有多少个病毒的氨基酸序列是致死的，其中 $mask$ 为长度为 L 的 $0/1/*$ 串。

输出格式

对于每次Q事件，输出一行一个整数，表示氨基酸序列是致死的病毒数量模 2^{32} 的结果。

样例一

input

```
4 8
Q ****
I 0101 5
I 1110 3
Q ****
Q *0**
Q ***0
I 0101 -3
Q *1**
```

output

```
0
8
0
3
5
```

explanation

第一天，培养皿中没有病毒，故致死的病毒数量为 0。

第二天，培养皿中新增了 5 个氨基酸序列为 `0101` 的病毒。

第三天，培养皿中新增了 3 个氨基酸序列为 `1110` 的病毒。

第四天，研究表明所有氨基酸序列都致死 ~~(WTF)~~，于是答案即为培养皿中病毒的总数 8。

第五天，所有满足第二个氨基酸为 0 (得氨酸) 的序列致死，而目前所有病毒的氨基酸序列中第二个氨基酸均为 1 (便氨酸)，故没有病毒致死。

第六天，所有满足第四个氨基酸为 0 (得氨酸) 的序列致死，于是这样的病毒共有 3 个，即氨基酸序列为 `1110` 的病毒。

第七天，由于某种原因，培养皿中氨基酸序列为 `0101` 的病毒死去了 3 个。

第八天，所有满足第二个氨基酸为 1 (便氨酸) 的序列致死，而目前所有病毒的氨基酸序列中第二个氨基酸均为 1 (便氨酸)，因此答案为病毒的总数 5。

样例二

见 "相关文件下载" 中的ex_amino2.in与 ex_amino2.out。

该组样例满足 $L=10,q=1000$，且保证 $mask$ 中不含 `1`。

样例三

见 "相关文件下载" 中的ex_amino3.in与ex_amino3.out。

该组样例满足 $L=10,q=1000$，且保证所有 `Q` 事件在 `I` 事件之后。

限制与约定

对于所有的测试点，均满足
$1\le L\le 18;1\le q\le 5\times 10^5;-10^9\le val\le 10^9;|seq|=|mask|=L$，且任意时刻每种氨基酸序列的病毒**数量非负**。

具体的子任务的数据规模见下图：

子任务	分值	L	q	其它性质
1	3	≤ 18	$\le 5\times 10^5$	保证没有 `Q` 事件
2	6			保证 $mask$ 中不含 `*`
3	12	≤ 10	≤ 1000	无
4	6	≤ 13	≤ 8000	
5	7	≤ 5	$\le 5\times 10^5$	
6	11	≤ 16	$\le 10^5$	保证 $mask$ 中不含 `1`
7	11			保证所有 `Q` 事件在 `I` 事件之后
8	8			无
9	13	≤ 18	$\le 5\times 10^5$	保证 $mask$ 中不含 `1`
10	13			保证所有 `Q` 事件在 `I` 事件之后
11	10			无

时间限制：3s

空间限制：512MB

反·易题

文件名：antieasy

题目描述

这是一道**提交答案题**。

小 γ 发现自己在 QYOJ 上的题被 Hack 了，非常气愤，于是准备找小 ω 理论。

小 ω 说：“你看看你，又把 `m` 定义成 `int` 类型了，你写一个 10^9 级别的单模 Hash 还不是随便卡？”

然而奇怪的是，在当时，小 γ 的程序通过了 QYOJ 上的大样例 —— 这也是小 γ 没有进行对拍的原因。

小 γ 向 QYOJ 的管理员要来了她的代码以及题目的数据 (QYOJ 不公开代码和数据)。可无奈的是，她在最终版的程序中的 $seed$ 是通过 `mt19937_64` 生成的。

形式化地，小 γ 的代码是如下的：

```
def m, b, p, q : int32
def gen : mt19937_64
def seed : uint64

function NEXT_INT()
    seed ← seed ⊕ seed ≫ 12
    seed ← seed ⊕ seed ≪ 25
    seed ← seed ⊕ seed ≫ 27
    return seed
end function

function RAND_INT(l, r)
    return NEXT_INT() mod (r − l + 1) + l
end function

procedure GENERATE()
    Randomize the Mersenne Twister generator gen          ▷ like gen.seed(time(NULL)); in C++
    seed ← Generate a pseudorandom number from gen        ▷ like seed = gen(); in C++
    repeat
        m ← UNIFORM_INT_DISTRIBUTION(3, 2^63 − 1, gen)    ▷ like m = std::uniform_int_distribution<uint64_t>(3, LLONG_MAX)(gen); in C++
    until IS_ODD_PRIME(m)
    for i ← 1 to 10 do
        NEXT_INT()
    end for
    b ← RAND_INT(1, m − 1)
    p ← RAND_INT(1, m − 1)
    q ← RAND_INT(1, m − 1)
end procedure

function W(c)
    if c = ( then
        return p
    else
        return q
    end if
end function

function HASH(s)
    return (∑_{i=1}^{|s|} W(s_i) · b^{i−1}) mod m
end function
```

其中 $\oplus, \ll, \gg$ 分别代表按位异或运算，无符号左移运算，无符号右移运算；函数 $\mathrm{Is\_Odd\_Prime}(m)$ 返回 true 当且仅当 m 是一个奇素数；字符串中的字符从 1 开始标号。

在 QYOJ 上，这个题有多组询问，每次询问一个 (不一定匹配的) 括号序列的两个长度相等的子串是否相同。

小 γ 会在主程序开头调用 Generate 函数，然后读入数据，对每组询问的两个子串分别调用 Hash 函数，得到两个值 H_1, H_2，若 $H_1 = H_2$ 则认为这两个串相同，否则认为其不相同。

要注意的一点是，你不需要担心上面这份代码是否会超时，小 γ 会对该算法进行快速的实现，但本质和上面的伪代码相同。

于是，小 γ 想要算一下，这个程序对询问的正确率评估如何，~~这样她就可以再次的感受到自己是一个多么欧的人子~~。然而时间已经过去了很多年，即使是小 γ 也没有办法完全回忆起当时自己的输出 (注意 QYOJ 和 SOJ 一样，是不存储用户输出文件的)。

幸运的是，她回忆起了大部分内容，即每个点的模数 m 和前一半询问的答案 (`Yes`/`No`)，唯一遗忘的是 $seed$ 和后一半询问的答案，于是她给了你输入数据和她回忆出的部分，想让你帮她还原 $seed$ 的**初值**和整个程序的输出，以便与管理员给她的标准答案进行 ~~`fc`~~ `diff`。

注意：你的任务是构造一份小 γ 程序的输出，而不是标准答案。小 γ 已经从管理员那儿得到了标准答案。

同时，小 γ 最终改正了错误，将这些变量重新定义回 64 位无符号整数类型。因此，你也需要对 $m < 2^{63}$ 的情形帮助小 γ。

输入格式

所有输入数据antieasy1.in~ antieasy20.in见数据下载，分别对应 20 个子任务。

第一行包含一个正整数 m，表示小 γ 回忆起的模数，保证 m 是一个奇素数。

第二行包含两个正整数 n, q，表示括号序列的长度和询问的个数。

第三行包含一个长度为 n 的，由 `(` 和 `)` 构成的字符串 s，描述整个括号序列。**注意括号序列不一定合法**。

接下来 q 行，每行三个整数 l, p_1, p_2，后面可选一个字符串。表示询问 $s\left[p_1..p_1+l-1\right]$ 与 $s\left[p_2..p_2+l-1\right]$ 是否相等，**注意字符从 1 开始标号**。

对于前 $\left\lceil \frac{q}{2} \right\rceil$ 行，后面跟着一个字符串 `Yes` 或 `No`，分别表示小 γ 的程序认为这两个串相同或不相同。

对于后 $\left\lfloor \frac{q}{2} \right\rfloor$ 行，后面没有其余信息。

输出格式

输出文件为antieasy1.out~antieasy20.out，分别对应相应的输入文件。

第一行输出一个整数，表示 $seed$ 的**初值**。你需要保证 $0 \leq seed \leq 2^{64}-1$。

接下来 q 行，每行一个为 `Yes` 或 `No` 的字符串，表示**小 γ 的原始输出**。

保证至少存在一组合法的解。如果有多组可能的输出，任意输出一组均可。

样例一

input

```
5
12 2
()(())((()))
6 1 7 Yes
4 3 8
```

output

```
38473274
Yes
Yes
```

样例二

input

```
5
12 2
()(())((()))
6 1 7 No
4 3 8
```

output

```
20041001
No
Yes
```

explanation

有可能不存在 Hash 冲突。

限制与约定

对于所有的测试点，保证
$3 \leq m \leq 2^{63}-25; 1 \leq n, q \leq 10^6; 1 \leq l \leq n; 1 \leq p_1, p_2 \leq n-l+1$，且 m 是素数。

评分方式

一共 20 个测试点，如果你输出的 $seed$ 可以产生正确的输出，且符合输入输出文件，则该测试点得 5 分，否则得 0 分。

小 ω 的魔方

文件名：rubik

题目描述

小 ω 在家里闷得慌，于是开始玩起了魔方。

在长时间的训练中，小 ω 对魔方已经玩腻了，不拘泥于只是还原魔方。于是她开始拆起了魔方。

与众不同的是，别人拆魔方都是拆那 26 个小块，她却开始拆那 54 张贴纸。

不一会儿，她就将原来魔方的 54 张贴纸拆完了。

正巧，小 χ 来到她家做客，看到小 ω 桌上~~琳琅满目的~~贴纸，也想自己组装一个魔方。

这些贴纸各不相同，有的画着可爱的猫猫，有的画着蜘蛛侠。

已知，桌上有 Y 种**黄色**贴纸、W 种白色贴纸、R 种**红色**贴纸、O 种**橙色**贴纸、B 种**蓝色**贴纸以及 G 种**绿色**贴纸。**每种颜色的贴纸均有无限多个**。

小 χ 对每种贴纸的喜好程度各有不同，具体地，(无视颜色的条件下) 她也将所有贴纸分别为三类：**好看的**，**一般般的**以及**难看的**，她分别将这三种贴纸的**好看度**定义为 $1, 0, -1$。

我们用 C_i ($C \in \{\mathtt{Y}, \mathtt{W}, \mathtt{R}, \mathtt{O}, \mathtt{B}, \mathtt{G}\}, i \in \{-1, 0, 1\}$) 表示颜色为 C 且好看度为 i 的贴纸种数，例如，B_{-1} 表示**难看的蓝色贴纸**的种数。

然后，她将一个魔方的好看度定义为所有 54 张贴纸的好看度之和。

由于小 χ 不懂魔方，因此她认为：**只要使用了每种颜色的贴纸各 9 张，所得到的就是一个合法的魔方**。尽管它可能不满足黄白相对，甚至无法复原，抑或是根本不存在 (比如黄白棱块就是不存在的)。

自然，小 χ 就有很多 "组装" 魔方的方案 ("组装" 即**贴贴纸**)。她想知道，对于每种最终的好看度 k，有多少种 (本质) 不同的 "组装" 魔方的方案，使得最终的好看度为 k 呢？

小 χ 瞥了一眼小 ω 的书房，发现小 ω 还有四阶、五阶，乃至 n ($n \leq 1000$) 阶魔方。于是她想对这些魔方都进行组装 ~~(反正她们待在家里，时间充裕)~~，于是你也需要帮她对这些高阶魔方进行回答。

当然，对于 n 阶魔方，上面的参数也需要进行更改。具体地，你需要将上文中的 $26, 54, 9$ 分别替换为 $6n^2 - 12n + 8, 6n^2, n^2$，特别地，当 $n = 1$ 时将它们替换为 $1, 6, 1$。

注意：如果两种方案可以在三维空间中旋转**整个魔方**而**全等**，则认为这两种方案是 (本质) 相同的。注意小 ω 和小 χ 生活在三维空间中，因此它们**不能对魔方进行镜像**。

两个方案**全等**，当且仅当每个位置上使用的贴纸种类相同，注意一种贴纸可以使用多次，你不需要关心贴纸的朝向问题，即不妨假设贴纸上的图案是完全对称的。

输入格式

第一行包含两个正整数 n, P，分别表示魔方的阶数以及输出控制参数。输出控制参数的具体含义将在「输出格式」中解释。

第二行包含三个非负整数 Y_{-1}, Y_0, Y_1，分别表示难看的，一般般的，好看的黄色贴纸的种数。

第三行包含三个非负整数 W_{-1}, W_0, W_1，意义同上。

第四行包含三个非负整数 R_{-1}, R_0, R_1，意义同上。

第五行包含三个非负整数 O_{-1}, O_0, O_1，意义同上。

第六行包含三个非负整数 B_{-1}, B_0, B_1，意义同上。

第七行包含三个非负整数 G_{-1}, G_0, G_1，意义同上。

输出格式

容易得到，最终魔方的好看度的取值范围为 $\left[-6n^2, 6n^2\right]$ 中的一个整数。**我们保证 P 是一个不超过 $12n^2+1$ 的正奇数**。

具体地，你只需要输出一行，包含 P 个整数，其中第 i ($1 \leq i \leq P$) 个整数表示**所有满足** $k \equiv i - \dfrac{P+1}{2} \pmod P$ **且** $k \in \left[-6n^2, 6n^2\right]$ 的答案 (即好看度为 k 的方案数) **模** 10^9+7 **后的异或和**。

注意：设置 P 仅仅是为了减少输出量大小，标准算法不依赖于 P 的取值。

样例一

input

```
1 1
0 1 0
0 1 0
0 1 0
0 1 0
0 1 0
0 1 0
```

output

```
30
```

explanation

特别地，一阶魔方无法进行 (除了整体外的) 旋转，于是它只是一个~~用于观赏的~~立方体。

所有颜色的贴纸只有一种，因此最终魔方的好看度一定为 0。

样例二

input

```
1 13
1 2 3
4 5 6
7 8 9
8 7 6
5 4 3
0 0 0
```

output

```
0 0 0 0 0 0 0 0 0 0 0 0 0
```

explanation

没有绿色贴纸，故无法组装成魔方。

样例三

input

```
2 1
0 1 0
0 1 0
0 1 0
0 1 0
0 1 0
0 1 0
```

output

```
940906141
```

explanation

精确的答案是 $135277941853080 = 2^3 \cdot 3^2 \cdot 5 \cdot 13 \cdot 131 \cdot 220653001$。

样例四

input

```
2 49
0 3 1
4 1 0
0 5 9
2 6 0
0 5 3
5 8 0
```

output

```
0 0 0 0 0 0 0 0 0 0 0 0 0 544749501 63051590 218701224 413209526 327036606
45871124 278106738 7308476 883834532 409278631 11668780 265789093
928895648 709317318 640274261 565809249 497467741 298018595 559073055
546178223 415655107 865831011 530349718 975815643 473116100 0 0 0 0 0 0 0
0 0 0 0 0
```

样例五

input

```
2 7
0 3 1
4 1 0
0 5 9
2 6 0
0 5 3
5 8 0
```

output

```
853739401 367219837 1039890180 359147035 571647095 900068407 155830037
```

explanation

该组样例为样例四的 "压缩" 后版本，因为避免过大的输出，常常会设置一个压缩系数。在本例是，七个数分别代表 $-3,-2,-1,0,1,2,3$ 这七个同余类。

如，输出的第五个数就表示所有好看度 $\equiv 1 \pmod 7$ 的方案数的异或和，即有 $571647095 = 0 \oplus 0 \oplus 278106738 \oplus 709317318 \oplus 415655107 \oplus 0 \oplus 0$。

样例六

input

```
36 13
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
16 17 18
```

output

```
656572069 316067026 492665298 350073114 109939730 141830812 721782114
395247388 823354680 250237378 268736953 334191590 721990568
```

样例七

见 "相关文件下载" 中的ex_rubik7.in与ex_rubik7.out。

该组样例满足子任务 25 的性质。

限制与约定

对于所有的测试点，均满足
$1 \leq n \leq 1000; 1 \leq P \leq \min\left\{12n^2+1, 200467\right\}; 0 \leq Y_{-1}, Y_0, Y_1, \cdots, G_{-1}, G_0, G_1 \leq 10^9+6$
，且 P 为奇数。

具体的子任务的数据规模见下表：

子任务	分值	n	其它性质
1	4	$=1$	S
2	3		D
3	3		无
4	3	$=2$	S
5	2		D
6	2		无
7	3	$=3$	S
8	2		D
9	2		无
10	4	≤ 10	S
11	3		D
12	3		M
13	2		无
14	4	≤ 35	S
15	3		D
16	3		M
17	2		无
18	5	≤ 200	S
19	4		D
20	4		M
21	3		无
22	6	≤ 1000 且 $\equiv \pm 1 \pmod 6$	S
23	5		D
24	5		M
25	4		无
26	5	≤ 1000	S
27	4		D
28	4		M
29	3		无

表中“其它性质”一栏，变量的含义如下：

- S (single)：对于所有颜色 C，保证 $C_{-1} = C_1 = 0$。
- D (double)：对于所有颜色 C，保证 $C_0 = 0$。
- M (monic)：对于所有颜色 C，保证 $C_{-1} = C_0 = C_1 = 1$。

提示：本题时限较紧，请注意实现常数 (放心标程两倍是有的)。

时间限制：4s

空间限制：1GB