**512MB,1S**

**红石计算机**

**问题描述**

[山谷村长]gbakkk5951并没有做过红石计算机，所以导致原型机漏洞百出，，他想通过输出数据是否能由某一组输入生成来更(dan)好(teng)地验证其正确性。已知输入有N个数（编号1到N）组成，和M次运算的结果,且所有数的二进制表示都有K位，gbakkk5951想知道是否存在一组输入数据使得所有的输出结果都被满足。

**输入描述**

第1行3个整数N,M,K

接下来M行，每行1个一整数A，一个运算符（一字节）op，一个整数B（中间有空格），一个空格，一个K位二进制整数C,表示第A个数op第B个数=C

若op为 ^ 表示异或

若op为 & 表示位与

若op为 | 表示位或

**输出描述**

若果(存在一组输入使得所有M个关系都被满足){

输出N行，每行一个二进制整数，表示第i号数的取值。（如果有多种方案 请任意输出一组）

}否则{

若果(存在最坏复杂度为多项式的2-SAT方案数统计算法){

输出一行Exist

}否则{

输出一行No

}

}

**样例输入**

【样例一】

3 3 3

1 ^ 2 000

3 & 3 101

2 ^ 1 000

【样例二】

3 3 3

1 ^ 2 001

1 & 2 110

3 & 1 001

【样例三】

2 2 3

1 | 2 111

1 & 2 000

【样例四】

10 20 5

6 & 9 00010

2 ^ 5 00001

9 & 4 01010

5 | 2 01111

5 ^ 5 00000

2 & 3 01000

2 ^ 8 11110

9 & 6 00010

4 ^ 8 01111

9 | 7 01110

1 ^ 7 11111

9 & 7 00000

10 | 9 11111

8 | 1 11011

1 | 4 11111

5 | 3 01111

7 & 8 00000

7 & 5 00100

3 ^ 2 00111

6 | 8 10110

**样例输出**

【样例一】

111

111

101

【样例二】

111

110

001

【样例三】

000

111

【样例四】

11011

01110

01001

11111

01111

00110

00100

10000

01010

10101

【样例五】（这不是输出）

若果(存在最坏复杂度为多项式的2-SAT方案数统计算法){

输出一行Exist

}否则{

输出一行No

}

**数据范围及提示**

前10%的数据K=1,0<=N<=20

前30%的数据K=1,0<=N<=50

前100%的数据0<=N<=500;1<=K<=150;0<=M<=10000;1<=A,B<=N

保证C为长度为K的二进制数；

样例只是提供了一种方案，若果存在多种方案，则输出任意一种即可（不要有多余输出），忽略行末空格和文末回车