

这天,早苗的面包又卖剩下了,秋生不得不再次去发面包.但是秋生是一个爱玩的人,已经对这枯燥的工作毫无兴趣了.于是他灵机一动,喊来了很多像春原这样的炮灰角色来打发无趣的工作.

随着秋生一声令下,春原军团们整整齐齐的站成了一个 $n*m$ 的矩形,然后秋生开始一个一个的发面包.然而,春原军团自然知道早苗面包的杀伤力,每个“春原”都想远离面包,珍爱生命.不过春原军团逃跑起来就像一堆无序运动的分子一样,他们只会左转,右转,有时候甚至被糟糕的面包吓软了腿.同时,春原的运动只和他站的位置有关,也就是说,站在 (i,j) 这个点的春原运动情况是固定的,无论是哪个春原都一样,我们用 G 来表示,取值是 L,R,N, G 矩阵的大小是 $(n-1)*(m-1)$ 的,下面会详细说明.

具体来说,春原们会进行如此运动:

首先,如果秋生当前在 (i,j) 点,那么 (i,j) 点上的春原会得到一个可怕的早苗面包...

然后,根据当前位置的 $G[i][j]$,来决定春原会怎么移动...

L:以 (i,j) 为左上角的 $2*2$ 的矩形逆时针旋转一格...

R:以 (i,j) 为左上角的 $2*2$ 的矩形顺时针旋转一格...

N:以 (i,j) 为左上角的 $2*2$ 的矩形保持不动...

一开始每个春原被依次编上 $1 \sim n*m$ 的编号,而秋生站在 $(1,1)$ 号点,每发完一个面包以后,秋生会向右移动一格,继续发面包.直到秋生在 $(1,m-1)$ 发完面包以后,他会到 $(2,1)$ 开始发面包.就是说,秋生发面包是按照先从左往右,再从上的顺序,如果到了 $(n-1)*(m-1)$ 号点还没有发完(因为是 $2*2$ 的矩形,所以只能发到 $(n-1)*(m-1)$),那么秋生会回到 $(1,1)$ 重新开始新一轮.

例如:

(1)	2	3	4	5	L	R	L	R
6	7	8	9	10	N	L	L	R
11	12	13	14	15	L	N	N	L
16	17	18	19	20				

(右边为 G)

2	(7)	3	4	5	2	6	(7)	4	5	2	6	4	(9)	5	2	6	4	3	9
1	6	8	9	10	1	8	3	9	10	1	8	7	3	10	(1)	8	7	10	5
11	12	13	14	15	11	12	13	14	15	11	12	13	14	15	11	12	13	14	15
16	17	18	19	20	16	17	18	19	20	16	17	18	19	20	16	17	18	19	20

(这是前几次的情况,分别是 1,7,7,9,1 号春原中枪...)

随着面包越发越久,春原们也是越来越怕,他们想知道到最后到底自己会拿到多少个早苗面包.但是因为早苗面包太难吃,所以卖剩下的自然也很多很多,多到春原们已经算不清这个数字了.于是你们懂得...

输入格式(clannad.in)

第一行有三个正整数 n,m,K , n 和 m 是春原军团的行数和列数, K 是卖剩的早苗面包的个数...接下来 $(n-1)*(m-1)$, 表示 G 矩阵,也就是春原们会怎么运动...

输出格式(clannad.out)

$n*m$ 行,每行一个数字,第 i 行的数字表示春原 i 号领到了多少个早苗面包...不过因为早苗卖剩的面包太多,你只需要输出 $\text{mod } 100000$ 以后的数字...

样例输入

3 3 10

LR

NL

样例输出

2

2

1

1

2

1

0

0

1

温馨提示

样例行列不同,不要读反行列丢掉该拿的分数哦...

不要因为 mod 错了一个数字而丢分...

时间限制可能将会因为评测机的问题而做出一些调整...

数据范围

[1,1],n,m≤10,K≤0

[2,3],n,m≤10,K≤100000

[4,9],n,m≤300,K≤10⁵⁰

[10,14],n,m≤300,K≤10⁷⁰⁰

[15,18],n,m≤500,K≤10²⁰⁰⁰⁰⁰

[19,20],n,m≤500,K≤10⁷⁰⁰⁰⁰⁰