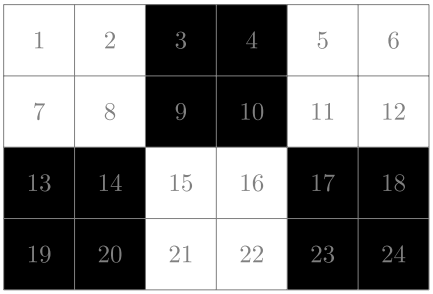
DMOPC '21 Contest 8 P4 - Grid Operations

Time limit: 2.0s **Memory limit:** 256M

You have found a strange device that shows a $2N \times 2M$ grid of numbers. The rows are numbered from 1 to $2N$, and the columns are numbered from 1 to $2M$. Initially, the number at the intersection of the i-th row and the j-th column is equal to $(i-1) \cdot 2M + j$.

The grid is partitioned into an $N \times M$ grid of 2×2 subgrids. These subgrids are coloured white or black in a checkerboard pattern, with the top-left 2×2 subgrid coloured white.



You notice that the device can perform 5 types of operations on the grid, which are:

- Type `1`: In every row, swap the numbers on each of the M adjacent pairs of cells with the same colour.
- Type `2`: In every row, swap the numbers on each of the $M-1$ adjacent pairs of cells with different colours.
- Type `3`: In every column, swap the numbers on each of the N adjacent pairs of cells with the same colour.
- Type `4`: In every column, swap the numbers on each of the $N-1$ adjacent pairs of cells with different colours.
- Type `5`: Rotate every white 2×2 subgrid clockwise and rotate every black 2×2 subgrid counterclockwise.

Now you wonder: What is the final state of the grid after performing Q operations of types $t_1, t_2, \ldots, t_Q$ in order?

Constraints

$1 \leq N, M, Q \leq 10^6$

$1 \leq N \times M \leq 10^6$

$1 \leq t_i \leq 5$

Subtask 1 [10%]

There are only type `1`, `3`, and `5` operations.

Subtask 2 [25%]

There are only type `1` and `2` operations.

Subtask 3 [10%]

There are only type `1`, `2`, `3`, and `4` operations.

Subtask 4 [55%]

No additional constraints.

Input Specification

The first line contains 3 space-separated integers: N, M, and Q.

The next Q lines each contain an integer t_i, the type of an operation to perform.

Output Specification

Output $2N$ lines, each containing $2M$ space-separated integers: The final grid after performing all the operations.

Sample Input

```
2 3 5
3
1
4
2
5
```

Sample Output

```
20 8 12 24 21 9
22 10 7 19 23 11
4 16 13 1 5 17
2 14 18 6 3 15
```

Explanation

After the first operation, the grid looks like:

7	8	9	10	11	12
1	2	3	4	5	6
19	20	21	22	23	24
13	14	15	16	17	18

After the second operation, the grid looks like:

8	7	10	9	12	11
2	1	4	3	6	5
20	19	22	21	24	23
14	13	16	15	18	17

After the third operation, the grid looks like:

8	7	10	9	12	11
20	19	22	21	24	23
2	1	4	3	6	5
14	13	16	15	18	17

After the fourth operation, the grid looks like:

8	10	7	12	9	11
20	22	19	24	21	23
2	4	1	6	3	5
14	16	13	18	15	17

After the fifth operation, the grid looks like:

20	8	12	24	21	9
22	10	7	19	23	11
4	16	13	1	5	17
2	14	18	6	3	15