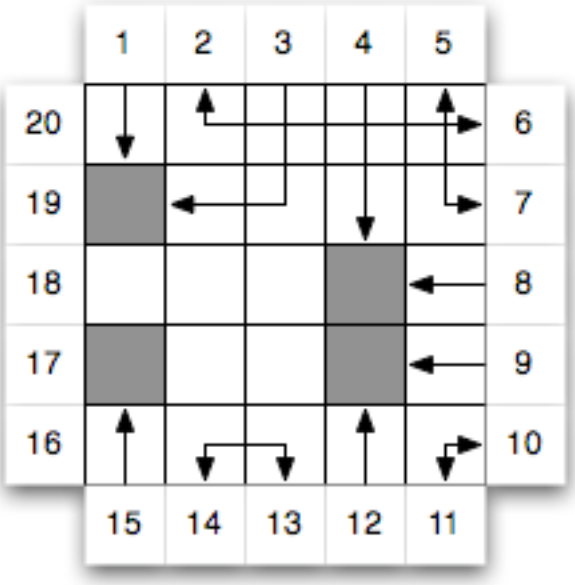
Problem 3: BlackBox

In Eric Solomon's game BlackBox a few marbles are hidden, each in one cell on a square grid. Your program should locate the marbles by shooting rays into the box.



The first line of input contains two positive numbers, the side length of the square and the number of hidden marbles (separated by white space and not necessarily equal); in this case:

```
5 4
```

The edges of the square are numbered, starting with 1, as shown above. To compute deflection, as discussed below, the edges are treated as if they were squares adjacent to the board. Board positions are identified by the ordered sequence of smallest edge numbers; for example, the top left gray box is at position `1,7`.

The remaining input lines describe what happens to search rays which enter the square from each side, proceed horizontally and/or vertically, and are either deflected or absorbed by the marbles.

A ray is absorbed if it hits a marble directly. A line describing an absorbed ray contains a single positive number identifying the edge position where the ray enters the square; in this case:

```
1
3
4
8
9
12
15
17
19
```

A ray is deflected if it hits one of the diagonal neighbors of a marble: it is turned 90 degrees away from the marble and from its current direction. A ray can be deflected more then once. A line describing a deflected ray contains two positive numbers identifying the edge positions where the ray enters and leaves the square (separated by white space); in this case:

```
2 6
5 7
10 11
13 14
18 18
```

Note that input and output edge of a ray can always be interchanged.

The last line is a special case: the ray starts at `18` and this edge is considered the diagonal neighbor of the marbles at `1,9` and `1,7`. The ray starts out going to the right. The first neighbor turns the ray up, the second neighbor turns the ray to the left. In summary, the ray is deflected twice and returns to its starting point `18`.

One would expect that the list of absorbed and deflected rays contains each edge at least once. However, this example shows that that need not be the case: A ray starting at `16` is turned from going right to going down and never even reaches the board. Similarly, a ray starting at `20` is immediately turned up and misses the board, too.

Your program should output one line with cell coordinates for each hidden marble. The coordinates should be the *smaller* edge numbers identifying column and row of the marble.

Sample Input

see `~/3.1`

```
5 5
1
2 6
3
4
5 7
8
9
10 11
12
13 14
15
17
18 18
19
```

Sample Output

```
1 7
4 8
1 9
4 9
```