3186   Make a Sequence

Your company's next product will be a new game, which is a three-dimensional variant of the classic game "Tic-Tac-Toe". Two players place balls in a three-dimensional space (board), and try to make a sequence of a certain length.

People believe that it is fun to play the game, but they still cannot fix the values of some parameters of the game. For example, what size of the board makes the game most exciting? Parameters currently under discussion are the board size (we call it n in the following) and the length of the sequence (m). In order to determine these parameter values, you are requested to write a computer simulator of the game.

You can see several snapshots of the game in Figures 3..5. These figures correspond to the three datasets given in the Sample Input.
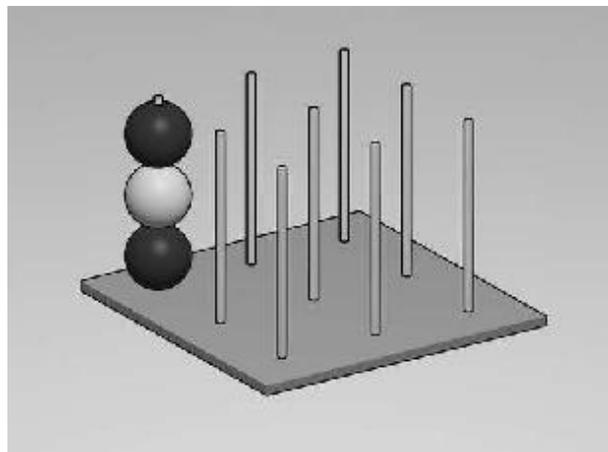
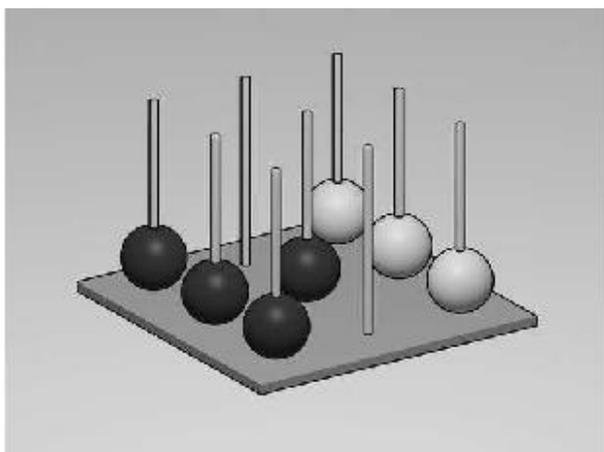

Figure 3: A game with $n = m = 3$

Figure 4: A game with $n = m = 3$
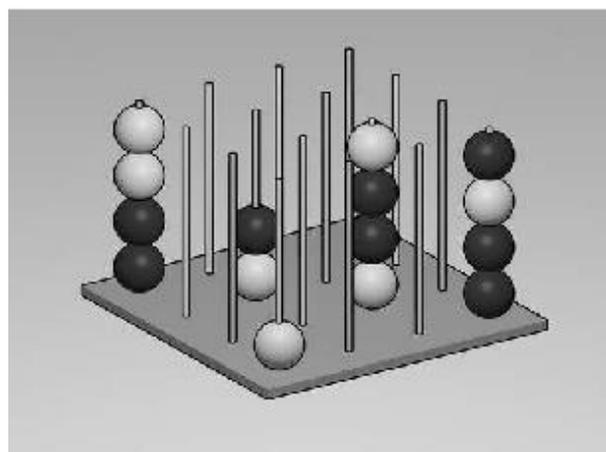(White made a 3-sequence before Black)

Figure 5: A game with $n = 4$, $m = 3$
(Black made two 4-sequences)

Here are the precise rules of the game.

1. Two players, Black and White, play alternately. Black plays first.

2. There are $n \times n$ vertical pegs. Each peg can accommodate up to n balls. A peg can be specified by its x- and y-coordinates ($1 \le x, y \le n$). A ball on a peg can be specified by its z-coordinate ($1 \le z \le n$). At the beginning of a game, there are no balls on any of the pegs.

3. On his turn, a player chooses one of $n \times n$ pegs, and puts a ball of his color onto the peg. The ball follows the law of gravity. That is, the ball stays just above the top-most ball on the same peg or on the floor (if there are no balls on the peg). Speaking differently, a player can choose x- and y-coordinates of the ball, but he cannot choose its z-coordinate.

4. The objective of the game is to make an *m*-sequence. If a player makes an *m*-sequence or longer of his color, he wins. An *m*-sequence is a row of *m* consecutive balls of the same color. For example, black balls in positions (5, 1, 2), (5, 2, 2) and (5, 3, 2) form a 3-sequence.

   A sequence can be horizontal, vertical, or diagonal. Precisely speaking, there are 13 possible directions to make a sequence, categorized as follows.

   (a) One-dimensional axes. For example, (3, 1, 2), (4, 1, 2) and (5, 1, 2) is a 3-sequence. There are three directions in this category.

   (b) Two-dimensional diagonals. For example, (2, 3, 1), (3, 3, 2) and (4, 3, 3) is a 3-sequence. There are six directions in this category.

   (c) Three-dimensional diagonals. For example, (5, 1, 3), (4, 2, 4) and (3, 3, 5) is a 3-sequence. There are four directions in this category.

   Note that we do not distinguish between opposite directions.

As the evaluation process of the game, people have been playing the game several times changing the parameter values. You are given the records of these games. It is your job to write a computer program which determines the winner of each recorded game.

Since it is difficult for a human to find three-dimensional sequences, players often do not notice the end of the game, and continue to play uselessly. In these cases, moves after the end of the game, i.e. after the winner is determined, should be ignored. For example, after a player won making an *m*-sequence, players may make additional *m*-sequences. In this case, all *m*-sequences but the first should be ignored, and the winner of the game is unchanged.

A game does not necessarily end with the victory of one of the players. If there are no pegs left to put a ball on, the game ends with a draw. Moreover, people may quit a game before making any *m*-sequence. In such cases also, the game ends with a draw.

Input

The input consists of multiple datasets each corresponding to the record of a game. A dataset starts with a line containing three positive integers n, m, and p separated by a space. The relations $3 \leq m \leq n \leq 7$ and $1 \leq p \leq n^3$ hold between them. n and m are the parameter values of the game as described above. p is the number of moves in the game.

The rest of the dataset is p lines each containing two positive integers x and y. Each of these lines describes a move, i.e. the player on turn puts his ball on the peg specified. You can assume that $1 \leq x \leq n$ and $1 \leq y \leq n$. You can also assume that at most n balls are put on a peg throughout a game.

The end of the input is indicated by a line with three zeros separated by a space.

Output

For each dataset, a line describing the winner and the number of moves until the game ends should be output. The winner is either 'Black' or 'White'. A single space should be inserted between the winner and the number of moves. No other extra characters are allowed in the output.

In case of a draw, the output line should be 'Draw'.

Sample Input

```
3 3 3
1 1
1 1
```

```
1 1
3 3 7
2 2
1 3
1 1
2 3
2 1
3 3
3 1
4 3 15
1 1
2 2
1 1
3 3
3 3
1 1
3 3
3 3
4 4
1 1
4 4
4 4
4 4
4 1
2 2
0 0 0
```

Sample Output

```
Draw
White 6
Black 15
```