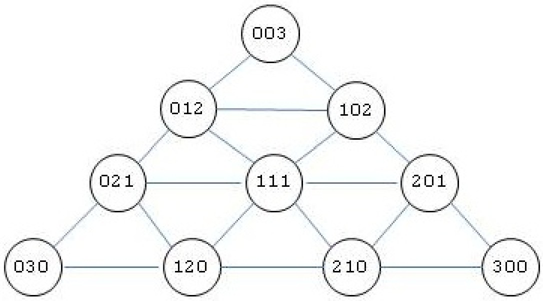
UVa Online Judge

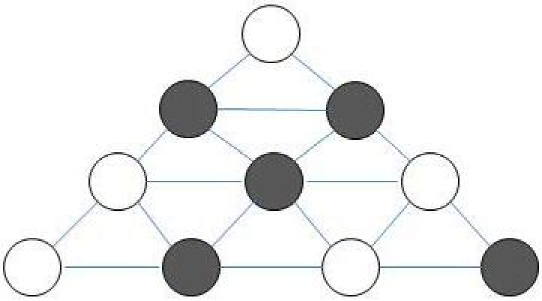
11841 Y-game

Willy and Benny enjoy very much playing *Y-game*! This is a game in which white and black tokens are placed on a triangular *n-grid*, $n \geq 0$, where n is called the order of the grid. A 3-grid is depicted in the figure below:



In general, an n-grid has $(n+2)(n+1)/2$ points with nonnegative "baricentric coordinates" (x, y, z), where $x + y + z = n$. Coordinates in a n-grid are assigned in such way that along right to left paths x-coodinates are constant, y-coordinates increase by one unit, and z-coordinates decrease by one unit (observe that this construction maintains $x + y + z = n$ true). Symmetric situations may be observed for left to right (where y-coordinates are constant) and horizontal (where z-coordinates are constant) paths. A point (x, y, z) in a n-grid is said to lay on the *x side* (resp., *y side*, *z side*) if and only if $x = 0$ (resp., $y = 0$, $z = 0$).

Willy uses white tokens and Benny uses black ones. Y-game rules are rather complicated, but the end of the game is attained when there is a token placed on every node of the grid. The winner is that player that has formed a Y, that is, his/her tokens are so placed that they include a connected set of points with a point on each side. For example, the following figure represents an end situation where Benny wins:



The winner is rather easy to determine when the grid is small. But Willy and Benny are not interested in that discussion today. Actually, they just want a software solution that computes the winner of ended Y-games. Could you help them?

Input

The problem input consists of several cases. A case begins with a line with two integer numbers, n and m, where n is the order of the grid and m the number of positions that have a black-coloured token (Benny's tokens), with $0 \leq n \leq 20$ and $0 \leq m \leq (n + 2)(n + 1)/2$.

Then, m lines follow, each one with 3 values x, y and z representing coordinate (x, y, z) of a point in the n-grid with a black token. Values on each input line are separated by one or more spaces.

The end of the input is signaled by a line

```
0 0
```

Output

Output texts for each input case are presented in the same order that the input is read. For an input case in the puzzle statement, the output should be a single line with the left-justified text

```
Willy
```

or

```
Benny
```

accordingly to the fact that Willy or, respectively, Benny wins in that case.

Sample Input

```
3 5
0 1 2
1 0 2
3 0 0
1 1 1
1 2 0
2 3
0 0 2
1 0 1
0 2 0
1 1
1 0 0
0 0
```

Sample Output

```
Benny
Willy
Willy
```