Injured Rook Problem

PROBLEM

Chess is a two-player board game believed to have been played in India as early as the sixth century. However, in this problem we will not discuss chess. I know you are familiar with the classic *n*-queen problem. That is, to plot *n* queens on an $n \times n$ chess board with the help of a classic backtracking algorithm. If you write that algorithm now you will find that there are 92 ways of plotting 8 queens on an 8×8 board provided no queens attack each other. However, in this problem we will not discuss the *n*-queen problem or its variations.

In this problem we will talk about rooks. A rook can move in either horizontal or vertical direction for any number of steps, as long as it does not fall off the chess board and no obstacle is blocking the way. However, an injured rook can only move in two directions – north and west. It cannot move south or move east.

Somebody has invented a two-player game called "Game of the Injured Rook". In this game, an injured rook is placed on an $N \times N$ chess board. Coordinates are assigned to each square as a (row, column) pair. The square at the northeast corner of the chess board is labelled as (1, 1). The square immediately east of it is labelled as (1, 2), and so on. The square (1, 1) is the destination square. There are some obstacles on the board, each one occupying exactly one square. An injured rook cannot pass through an obstacle. Initially an injured rook is placed on an empty square. The game then starts.

The game is played by two people. For convenience we call them Man and Nang. The two players take turns to move the injured rook. Only one move is allowed in one turn. The game ends when the injured rook reaches the destination square, or one of the players cannot move the injured rook. The loser is the one who is unable to move the injured rook. The winner is the one who moves the injured rook to the destination square, or the one who is not the loser.

It can be shown that for any scenario, either one of the player has a winning strategy. That means he can always win no matter how the other player plays, by following a certain strategy (the strategy may be very complex). You are asked to write a program that, given a scenario, determines which player has a winning strategy. Assume that Man always starts first.

INPUT

The first line of the input contains an integer N, $1 \le N \le 1000$, the size of the chess board. The next line contains an integer M, $0 \le M \le 100000$, the number of obstacles. Each of the following M lines contains the row number followed by the column number of an obstacle. The row numbers and column numbers are positive integers

not greater than *N*. The last line contains a pair of integers *r* and *c*, $1 \le r,c \le N$, the starting row number and the column number of the injured rook. It is guaranteed that the starting position of the injured rook is on an empty square.

OUTPUT

The output consists of only one line which contains the name of the player who has a winning strategy. If Man has a winning strategy, output ``Man'' (without the quotes), otherwise output ``Nang'' (without the quotes).

EXAMPLE INPUT AND OUTPUT

Example 1:

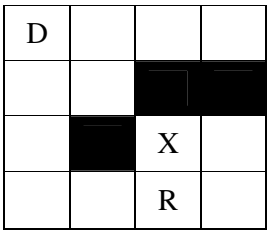
input

```
4
3
2 3
2 4
3 2
4 3
```

output

```
Man
```

The following figure shows the scenario for the input given above. D is the destination square, R is the injured rook and the shaded squares are obstacles.



When the game starts, Man moves the injured rook one step north to X. Nang cannot move the injured rook and so Man is the winner.