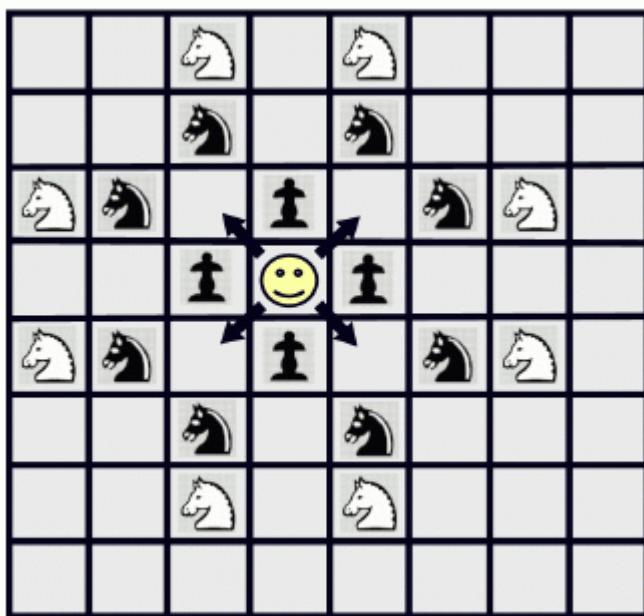


## 10477 The Hybrid Knight

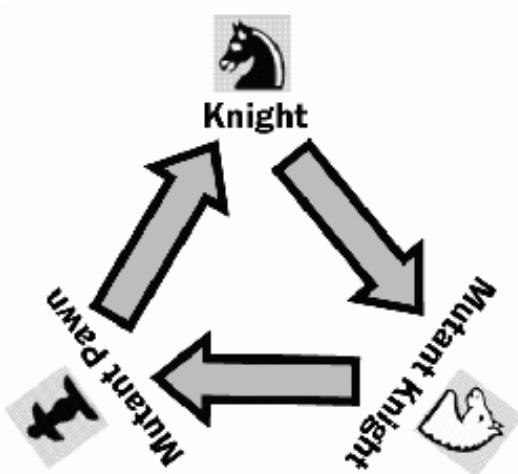
Hybridization is a very important concept in Biology, Chemistry and in many other fields. But is not at all an important concept in chess. Today we shall introduce a new chess knight, which can move like a normal knight, a mutant knight and like a mutant pawn in a chessboard.

The following figure illustrates the different types of moves of our hybrid knight. Please note that the knight is never allowed to land outside the board.



The figure on the left shows that our hybrid knight is placed at the position of the smiling face. If its initial movement is like a knight it can reach the places marked with the knight, If its initial movement is like a mutant knight it can reach the places marked with the mutant knight. The places marked with mutant pawns have similar meanings. However if the final move is made with the mutant pawn, the pawn has the option to use its attacking moves. In the attacking move of the mutant pawn, it can move one unit diagonally.

- = Knight
- = Mutant Knight
- = Mutant Pawn
- = Mutant Pawn's optional Attacking Directions for the final move



corner is 64.

Unfortunately, the knight cannot move according to his own will. It maintains the following cyclic order when it chooses its movement. It means if the knight's first movement is like a knight, the second movement will be like a mutant knight, the third movement will be like a mutant pawn, the fourth movement will be again like a knight and so on. The same cycle applies if the hybrid knight starts with the mutant knight move or with the mutant pawn move. The cycle to be followed is shown in the given figure.

Given the size of the square shaped board, your job is to find the minimum number of moves required by the hybrid knight to reach from one board position to another. The board positions are numbered in row major order from top to bottom. So in a  $(8 \times 8)$  chessboard the identifier of the top-left corner is 1 and the identifier of the bottom right

### Input

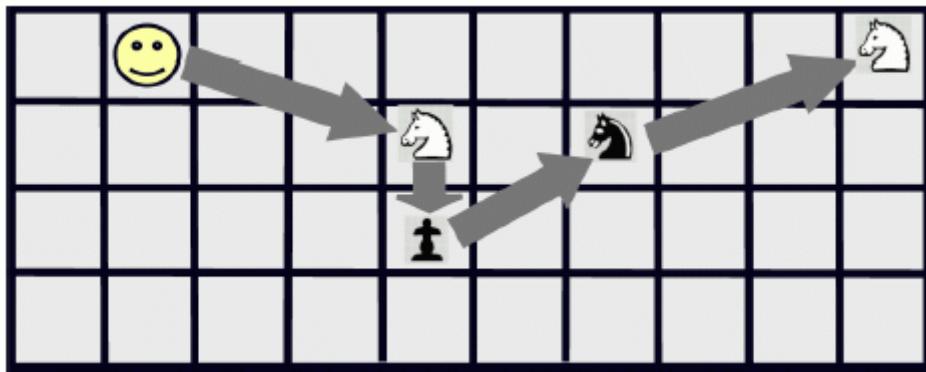
The input file contains at most 20 sets of input. Each set starts with two integers  $N$  ( $4 \leq N \leq 20$ ) and  $S$  ( $S \leq 1000$ ).  $N$  indicates that you have to work with a  $N \times N$  board and  $S$  is the total number

of queries in that set. Each of the next  $S$  lines contains 1 query. Each query contains two integers  $B$  and  $E$  ( $1 \leq B, E \leq N * N$ ). Here  $B$  is the starting location of the knight and  $E$  is the destination location of the knight. Input is terminated with a set whose value of  $N$  is zero. This case should not be processed. You can assume that the hybrid knight takes 0 moves to go from X to X, where X is a valid board position. However, there will be no such query where  $B$  and  $E$  is equal.

## Output

For each set of input you should output the serial of the set as shown in the output for sample input. For each query in a set you should output the minimum number of moves required by the hybrid knight to move from  $B$  to  $E$ . If the destination is unreachable from the source, print a ‘?’ without the quotes.

**Note:** The illustration given below shows the movement required for set 2. Here to go from 2 to 10 in a  $10 \times 10$  board the hybrid knight uses its mutant knight move first to go to cell number 15. Then it uses the mutant pawn move to go one row down to cell number 25. The third move is like a regular knight, it takes our knight to cell number 17. The fourth and final move is a mutant knight move that takes our knight to the destination. We’re showing the first 4 rows of the  $10 \times 10$  board here.



## Sample Input

```
5 2
10 15
10 20
10 1
2 10
0 10
```

## Sample Output

```
Set 1:
1
2
Set 2:
4
```