Special Case: $\mathbf{M = 1}$

There is no positive divisor of 1 which is smaller than 1, so there is no possible move to be made when $M = 1$. Because of this, player 2 will always win in this scenario.

When $\mathbf{N}$ is Even

Imagine that the towers are separated into two groups having an equal number of $\frac{N}{2}$ towers in each group. There is a 1 to 1 relationship between these two groups; whenever player 1mutates one of the towers from the first group, player 2 can simply copy player 1's last move and apply it to a tower from the second group.

In this way, player 2 will always have move to make (i.e., the mirror of player 1's previous move), so player 2 will always win.

When $\mathbf{N}$ is odd

Player 1 choses a tower and breaks it down to a height of 1. This results in $N - 1$ remaining breakable towers, which is an even number. Because we know that the first player to make a move when there are an even number of towers always loses (see above, when N is even), we know that player 1 will always win.


Set by forthright48

Problem Setter's code :

C++

```
#include <bits/stdc++.h>
using namespace std;
void solution() {
    int kase;
    scanf ( "%d", &kase );
    while ( kase-- ) {
        int n, m;
        scanf ( "%d %d", &n, &m );
        ///If m = 1, then there is nothing to do. Player 2 always wins
        if ( m == 1 || n % 2 == 0 ) {
            printf ( "2\n" );
        }
        else printf ( "1\n" );
    }
}
```

```
int main () {

    solution();

    return 0;
}
```