Task From Mr X

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 second
Memory limit:	1024 mebibytes

Mr. X has finally decided to contribute to computer science by inventing a new data structure. After many nights of thinking, he has decided that the data structure will manage an array A of size 2^n (0-indexed). All of it's entries will be 0 initially. This array will be called **A: version 0**. Note that the data structure will maintain multiple versions of the arrays.

The thing is, Mr. X can't decide how to implement this elusive data structure. It's main two operations are:

1. User will give 3 integers i x v, which means that take A: version v and consider all indices j. If (i & $j == j$) then assign x to the element at index j. Here & is the bitwise and operation. This process will create a new version of A with the changed entries. If the latest created version is w then this operation will create version $w + 1$. **Note that version v is not changed**
2. User will give 2 integers i v, which means that take A: version v and output the entry A[i] from that.

The v in both operation will be an **already created version**.

Now suddenly Mr. X remembers that he has yet to finish his data structure and algorithms course in university! So he decides to do the obvious thing, he tasks you, a high school student, to implement this data structure for him. So, help him, you're his last hope!

Input

The first line will contain two integers n and q, where n is defined above and q is the number of queries to be performed. Next q lines will describe the two kinds of queries. The first type of the query is described as 1 i x v and second kind is described as 2 i v, where i, x, v are integers.

Output

For each query of the second type, output the number at index i in a single line.

Scoring

For all tests $0 \le x \le 10^9$ and $0 \le i < 2^n$. And for some query, if versions $0, 1, \cdots, w$ are created so far,then $0 \le v \le w$.

- Subtask 1 (9 points): $0 \le n \le 12, 1 \le q \le 1000$, for each query, v will be the **latest created version** of A.
- Subtask 2 (17 points): $0 \le n \le 17, 1 \le q \le 100000$, for each query, v will be the **latest created version** of A, each i in update operation will be **distinct.**
- Subtask 3 (53 points): $0 \le n \le 17, 1 \le q \le 100000$, for each query, v will be the **latest created version** of A.
- Subtask 4 (21 points): $0 \le n \le 17, 1 \le q \le 100000$

Example

standard input	standard output

2 6
1 2 1 0
1 0 2 1
2 3 2
2 2 2
2 1 2
2 0 2
	0
	1
	0
	2

Explanation

- Here in the end, the data structure has 3 versions of A. **version 0**: [0, 0, 0, 0], **version 1**: [1, 0, 1, 0], **version 2**: [2, 0, 1, 0]. So querying on version 2 gives the above results.