# Problem C. Cumulative Code

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 7 seconds |
| Memory limit: | 512 mebibytes |

As you probably know, a *tree* is a graph consisting of $n$ nodes and $n-1$ undirected edges in which any two nodes are connected by exactly one path. In a *labeled tree* each node is labeled with a different integer between 1 and $n$.

The *Prüfer code* of a labeled tree is a unique sequence associated with the tree, generated by repeatedly removing nodes from the tree until only two nodes remain. More precisely, in each step we remove the *leaf* with the smallest label and append the label of its *neighbour* to the end of the code. Recall, a leaf is a node with exactly one neighbour. Therefore, the Prüfer code of a labeled tree is an integer sequence of length $n-2$. It can be shown that the original tree can be easily reconstructed from its Prüfer code.

The *complete binary tree of depth* $k$, denoted with $C_k$, is a labeled tree with $2^k - 1$ nodes where node $j$ is connected to nodes $2j$ and $2j+1$ for all $j$ ¡ $2^{k-1}$. Denote the Prüfer code of $C_k$ with $p_1, p_2, \ldots, p_{2^k-3}$.

Since the Prüfer code of $C_k$ can be quite long, you do not have to print it out. Instead, you need to answer $n$ questions about the sums of certain elements on the code. Each question consists of three integers: $a$, $d$ and $m$. The answer is the sum of the of the $C_k$'s Prüfer code elements $p_a$, $p_{a+d}$, $p_{a+2d}$, $\ldots$, $p_{a+(m-1)d}$.

## Input

The first line contains two integers $k$ and $q$ ($2 \le k \le 30$, $1 \le q \le 300$) — the depth of the complete binary tree and the number of questions. The $j$-th of the following $q$ lines contains the $j$-th question: three positive integers $a_j$, $d_j$ and $m_j$ such that $a_j$, $d_j$ and $a_j + (m_j - 1)d_j$ are all at most $2^k - 3$.

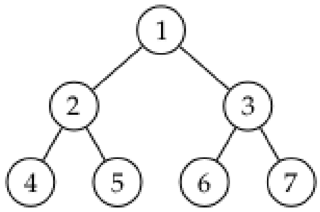
## Output

Output 1 lines. The $j$-th line should contain a single integer — the answer to the $j$-th question.

## Example

| standard input | standard output |
|---|---|
| 3 5<br>1 1 1<br>2 1 1<br>3 1 1<br>4 1 1<br>5 1 1 | 2<br>2<br>1<br>3<br>3 |
| 4 4<br>2 1 5<br>4 4 3<br>4 8 1<br>10 3 2 | 18<br>15<br>5<br>13 |
| 7 1<br>1 1 125 | 4031 |

## Note




In the first example above, when constructing the Prüfer code for $C_3$ the nodes are removed in the following order: 4, 5, 2, 1, 6. Therefore, the Prüfer code of $C_3$ is 2, 2, 1, 3, 3.