Problem:

Create a list, , of empty sequences, where each sequence is indexed from to . The elements within each of the sequences also use -indexing.

- Create an integer, , and initialize it to .
- The types of queries that can be performed on your list of sequences () are described below:
  1. Query: `1 x y`
     1. Find the sequence, , at index in .
     2. Append integer to sequence .
  2. Query: `2 x y`
     1. Find the sequence, , at index in .
     2. Find the value of element in (where is the size of) and assign it to .
     3. Print the new value of on a new line

Task

Given , , and queries, execute each query.

Note: is the *bitwise XOR* operation, which corresponds to the `^` operator in most languages. Learn more about it on Wikipedia.

Input Format

The first line contains two space-separated integers, (the number of sequences) and (the number of queries), respectively.

Each of the subsequent lines contains a query in the format defined above.

Constraints

-
-
-
- It is guaranteed that query type will never query an empty sequence or index.

Output Format

For each type query, print the updated value of on a new line.

Sample Input

```
2 5
1 0 5
1 1 7
1 0 3
2 1 0
2 1 1
```

Sample Output

```
7
```

```
3
```

Explanation

Initial Values:

 = [ ]
 = [ ]
Query 0: Append to sequence .

 = [5]
 = [ ]
Query 1: Append to sequence .
 = [5]
 = [7]
Query 2: Append to sequence .

 = [5, 3]
 = [7]
Query 3: Assign the value at index of sequence to , print .

 = [5, 3]
 = [7]
```
7
```
Query 4: Assign the value at index of sequence to , print .

 = [5, 3]
 = [7]
```
3
```

Solution:

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
```

```
int main() {
    /* Enter your code here. Read input from STDIN. Print
output to STDOUT */
    //definign the sequences
    int n, num;
    int a, b,c;
    cin>>n>>num;
    vector <vector <int>> array(n);
    //int count[n]={0};
   // int num=0; //holds down the number of queries to be
worked on
    int last_answer=0;
    int size=array.size();

    for( int i=0; i<num; i++)
     { cin>>a>>b>>c;
       if(a==1)
            { //evaluate the expression to find the sequence
to use for append operation
             //cout<<endl;
             int index=(b ^ last_answer)%n;
            //cout<<a<<": "<<" index: "<<index;
                array[index].push_back(c);
                //count[index]++;
             }
       else if(a==2)
             {  //cout<<endl;
                int index=(b ^ last_answer) % n;
            //cout<<"a: "<<a<<" index: "<<index;
            //cout<<"size: "<<size;
                last_answer=array[index][c%
(array[index].size()) ];
                cout<<last_answer<<endl;

              }
     }

    return 0;
}
```