Super Borboletas World

Super Borboleta World

The input/output were broken, I've fixed it and rejudged all submissions.

Raphaell is a well-known programmer who created the biggest game development company in the world, BGM (Boboleta's GameMaker). As recently one of its game – S.B.W (Super Borboleta's World) - has became very popular, Raphaell decided to make an online version of S.B.W's game. In order to do this he'll expose the source code and the mechanism of that game so anyone is able to improve it.

At first the game is made of three main operations in which the user is able to call as much as necessary. As the game is composed by K arrays of lists where each list has at most N integers on it, the three operations can be described in the following way:

- Operation <2> <x> <y>: Insert the integer <y> to the end of the <x>-th list.

- Operation <1> <x> <y>: Clean every list whose index lie on the range between <x> and <y> (inclusive).

- Operation <0> <x> <y>: In each list between <x> and <y> calculate all the possible consecutive XOR sum's, where XOR stands for the operation Exclusive OR, and return the maximum value of all possible XOR sum's.

Raphaell has access to the original pseudocode which is given bellow:

```
m <- array( array() )

def insert(x,y):
    insert y to m[x]

def clear(x,y):
    for i<-x to y:
        clear m[i]

def max_xor(x,y):
    best <- 0
    for i<-0 to sizeOf m[x]:
        sum_xor <- 0
        for j<-i to sizeOf m[x]:
            sum_xor <- sum_xor (xor) m[x][j]
            best <- max(best,sum_xor)
    if x < y:
        best <- max(best, max_xor(x+1,y))
    return best
```

This implementation was efficient to the offline version of the game. However, as the online version may receive a thousands of players at once, it's necessary many optimizations to run the game properly. Even though his friend has already tried really hard to figure a way to improve the performance, he hasn't got any good results till now.

Input

The input contains several test cases. A test case begins with a line containing an integer Q (1 ≤ Q ≤ 10^5), where Q represents the number of operations that are going to be performed. Then follow Q lines, each containing an operation. All the operations are as described above:

- 0 x y: In each list between x and y calculate all the possible consecutive XOR sum's and return the maximum possible value.

- 1 x y: Clean every list whose index lie on the range between x and y inclusive.

- 2 x y: Insert the integer y to the end of the x-th list.

Both x and y in every operation will never exceed 10^14. The last test case is followed by a line containing a single 0.

Output

For each query <0> <x> <y> print a line containing a single integer representing the maximum possible XOR as described above.

Example

Input	Output
14	3
2 2 1	7
2 2 2	3
2 2 1	0
2 2 1	
2 2 2	
2 3 1	
2 3 2	
2 3 7	
0 1 2	
0 2 3	
1 3 3	
0 1 3	
1 1 3	
0 1 3	
0	