

Mirko has received an interesting homework assignment: to design his own little processor (Mirkoprocessor). The processor has  $N$  registers with indices from 1 to  $N$ , and each register holds one unsigned 32-bit integer in the usual binary format (the possible values range from 0 to  $2^{32} - 1$ ).

The processor is capable of executing the following instruction types:

Instruction type	Description	Example
1 $K M$	Rotate the bits of register $K$ by $M$ positions to the right; write the result back to register $K$ .	<pre>000000000000000000010001111111011       → (M = 1010) → 111111101100000000000000000001000 (in base 10: 9211 → (M = 10) → 4 273 995 784)</pre>
2 $K L$	Compute the bitwise XOR of registers $K$ and $L$ ; output the result to the system bus.	<pre>00000000000000000000000001111000111 XOR 00000000000001111100000000000111 = 00000000000001111100001111000000 (in base 10: 967 XOR 507 911 = 508 864)</pre>

Mirko has already built a model of the processor, and only then realized that he'd forgotten to include an operation to read the contents of a register. Now, his only option is to execute a large number of type 1 and type 2 instructions and infer the register contents from the results. Having executed a sequence of commands, he has asked you to help him derive the initial register contents consistent with the obtained results.

If there are multiple possibilities for the initial register state combination, find the **lexicographically smallest** one. (If two combinations have equal values in the first  $K - 1$  registers and different values in register  $K$ , the lexicographically smaller combination is the one with the smaller value in register  $K$ .)

### INPUT

The first line of input contains two positive integers:  $N$  ( $2 \leq N \leq 100\,000$ ), the number of registers, and  $E$  ( $1 \leq E \leq 100\,000$ ), the number of executed instructions.

The remaining input lines describe the instructions in the order that they were executed by Mirkoprocessor, formatted as described in the table above, one per line. All instructions are legal (the following conditions hold:  $1 \leq K, L \leq N$ ,  $0 \leq M < 32$ ). Each instruction of **type 2** is followed by another line containing a positive integer between 0 and  $2^{32} - 1$ , inclusive – the result of that operation (the bitwise XOR value) in base 10.

---

---

**OUTPUT**

The first and only line of output must contain the required **N** register values, separated by spaces.

If there is no possible combination of initial values consistent with input, output only the number -1, to notify Mirko that his processor has a bug.

**SCORING**

In test data worth a total of 64 points, the numbers **N** and **E** will be smaller than 1000.

**SAMPLE TESTS**

<b>input</b> 3 3 2 1 2 1 2 1 3 2 2 2 3 3	<b>input</b> 4 6 2 4 2 3 2 4 1 6 1 3 1 2 3 1 2 1 2 2 2 2 3 7	<b>input</b> 5 6 2 4 2 10 2 5 3 2 2 2 3 1 2 1 4 3 1 3 1 2 3 4 2147483663
<b>output</b> 0 1 2	<b>output</b> 5 0 14 3	<b>output</b> 15 6 7 12 5