A. Levko and Array Recovery

Levko loves array $a_1, a_2, \ldots, a_n$, consisting of integers, very much. That is why Levko is playing with array a, performing all sorts of operations with it. Each operation Levko performs is of one of two types:

1. Increase all elements from l_i to r_i by d_i. In other words, perform assignments $a_j = a_j + d_i$ for all j that meet the inequation $l_i \leq j \leq r_i$.
2. Find the maximum of elements from l_i to r_i. That is, calculate the value .

Sadly, Levko has recently lost his array. Fortunately, Levko has records of all operations he has performed on array a. Help Levko, given the operation records, find at least one suitable array. The results of all operations for the given array must coincide with the record results. Levko clearly remembers that all numbers in his array didn't exceed 10^9 in their absolute value, so he asks you to find such an array.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 5000$) — the size of the array and the number of operations in Levko's records, correspondingly.

Next m lines describe the operations, the i-th line describes the i-th operation. The first integer in the i-th line is integer t_i ($1 \leq t_i \leq 2$) that describes the operation type. If $t_i = 1$, then it is followed by three integers l_i, r_i and d_i ($1 \leq l_i \leq r_i \leq n$, $-10^4 \leq d_i \leq 10^4$) — the description of the operation of the first type. If $t_i = 2$, then it is followed by three integers l_i, r_i and m_i ($1 \leq l_i \leq r_i \leq n$, $-5 \cdot 10^7 \leq m_i \leq 5 \cdot 10^7$) — the description of the operation of the second type.

The operations are given in the order Levko performed them on his array.

Output

In the first line print "YES" (without the quotes), if the solution exists and "NO" (without the quotes) otherwise.

If the solution exists, then on the second line print n integers $a_1, a_2, \ldots, a_n$ ($|a_i| \leq 10^9$) — the recovered array.

Examples

input
4 5
1 2 3 1
2 1 2 8
2 3 4 7
1 1 3 3
2 3 4 8

output
YES
4 7 4 7

input
4 5
1 2 3 1
2 1 2 8
2 3 4 7
1 1 3 3
2 3 4 13

output
NO