C. Sereja and Prefixes

Sereja loves number sequences very much. That's why he decided to make himself a new one following a certain algorithm.

Sereja takes a blank piece of paper. Then he starts writing out the sequence in m stages. Each time he either adds a new number to the end of the sequence or takes l first elements of the current sequence and adds them c times to the end. More formally, if we represent the current sequence as $a_1, a_2, ..., a_n$, then after we apply the described operation, the sequence transforms into $a_1, a_2, ..., a_n[, a_1, a_2, ..., a_l]$ (the block in the square brackets must be repeated c times).

A day has passed and Sereja has completed the sequence. He wonders what are the values of some of its elements. Help Sereja.

Input

The first line contains integer m ($1 \leq m \leq 10^5$) — the number of stages to build a sequence.

Next m lines contain the description of the stages in the order they follow. The first number in the line is a type of stage (1 or 2). Type 1 means adding one number to the end of the sequence, in this case the line contains integer x_i ($1 \leq x_i \leq 10^5$) — the number to add. Type 2 means copying a prefix of length l_i to the end c_i times, in this case the line further contains two integers l_i, c_i ($1 \leq l_i \leq 10^5$, $1 \leq c_i \leq 10^4$), l_i is the length of the prefix, c_i is the number of copyings. It is guaranteed that the length of prefix l_i is never larger than the current length of the sequence.

The next line contains integer n ($1 \leq n \leq 10^5$) — the number of elements Sereja is interested in. The next line contains the numbers of elements of the final sequence Sereja is interested in. The numbers are given in the strictly increasing order. It is guaranteed that all numbers are strictly larger than zero and do not exceed the length of the resulting sequence. Consider the elements of the final sequence numbered starting from 1 from the beginning to the end of the sequence.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print the elements that Sereja is interested in, in the order in which their numbers occur in the input.

Examples

input
6 1 1 1 2 2 2 1 1 3 2 5 2 1 4 16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

output
1 2 1 2 3 1 2 1 2 3 1 2 1 2 3 4