Data Structure

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	512 megabytes

Andy is a famous data structure expert at Nanjing University second to none. One day he throws a plain dry data structure problem to his friends, but none of them can solve. How about you?

Given a tree rooted at node 1. Each node has a weight which is 0 initially. Define the distance between two nodes as the number of edges in the unique simple path between the two nodes. You need to perform these two types of operations:

- Type 1: given a, x, y, z, add z to the weights of all a's descendants (including a itself) whose distances to a are y modulo x;
- Type 2: given a, return the weight of node a.

Input

The first line of the input is a single integer T ($1 \leq T \leq 4$), the number of test cases.

Each test cases starts with two integers n, m ($1 \leq n, m \leq 300000$), denoting that there are n nodes (numbered 1 through n) in the tree and you need to perform m operations. The next line contains $n-1$ integers, $f_1, f_2, \cdots, f_{n-1}$ ($1 \leq f_i \leq i$), specifying the edges of the trees; the ith integer denotes the parent of node $i+1$. The next m lines describe the operations. Each line is either `1 a x y z` ($1 \leq a \leq n, 1 \leq x \leq n, 0 \leq y < x, 0 \leq z \leq 500$), denoting an operation of type 1, or `2 a` ($1 \leq a \leq n$), denoting an operation of type 2.

Output

For each operation of type 2 in each test case, print the answer in one line.

Example

standard input	standard output
1 5 5 1 1 2 1 1 1 5 4 1 1 1 4 1 5 1 2 1 0 4 2 3 2 1	5 0