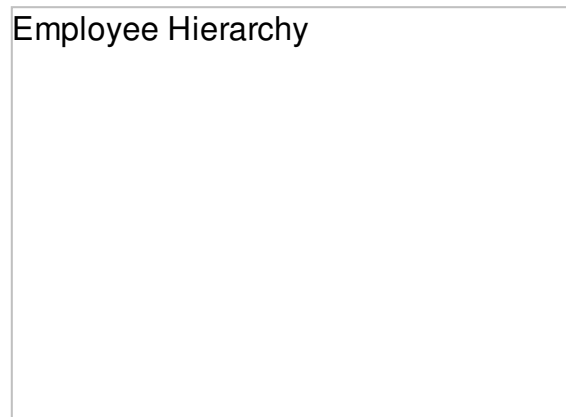


Salary Management

You are working as a software engineer of Moogle! Now the boss of Moogle wants you to create a program which will efficiently handle some operations. At first, we should tell you about the structure of the employees at Moogle! Each employee has an integer id from 1 to N. Where 1 is the id of the Managing Director of Moogle! , who is the greatest boss of all the employees. Then except the Managing Director, each employee has an immediate boss. No employees have more than one immediate boss. Now your boss wants to have some operation like this:

1. He will give you an ID of an employee; you need to find the sum of salary of all the employees under that employee including him/herself.
2. He will give you an ID of an employee; you need to increase the salary of all the employees under that employee including him/herself by the minimum of minimum salary of all the employees under that employee including him/her and 1000.

Let's see the structure, hope you will get a clear idea about the problem.



Employee Hierarchy

Salary Table:

ID	1	2	3	4	5	6	7
Salary (BDT)	500	300	200	100	10	200	100

Now if your boss wants to do the first operation for Employee ID 2. Then the output will be: $300 + 100 + 200 = 600$ BDT.

If your boss wants to do the second operation for the Employee ID 1, then the salary table becomes:

ID	1	2	3	4	5	6	7
Salary (BDT)	510	310	210	110	20	210	110

As minimum salary is 10 for Employee id 5 and which is less than 1000.

Input

Input starts with an integer T (≤ 3), denoting the number of test cases.

The first line of a case is a blank line. The next line contains two integers N ($1 \leq N \leq 10^5$), q ($1 \leq q \leq 50000$) where N denotes the number of nodes and q denotes the number of queries. The nodes are numbered from 1 to N .

Then there will be N lines. The i^{th} ($1 \leq i \leq N$) line contains two integers p_i and s_i ($0 \leq p_i \leq N$, $1 \leq s_i < 500$). p_i denotes the parent and s_i denotes the salary of the i^{th} employee, respectively. You can assume that the employee id with 1 is the managing director and only its parent is 0.

Each of the next q lines contains a query. Each query contains two integers: c and v ($1 \leq c \leq 2$, $1 \leq v \leq N$), where c denotes operation type, and v denotes the employee id. If the $c = 1$, then it means to do the first operation. If $c = 2$, then the second operation.

You can assume that the input builds a valid rooted tree.

Output

For each case, print the case number in a line. Then for each query type 1, print the sum of the salary.

Example

Input:

1

7 3
0 500
1 300
1 200
1 100
3 10
2 200
2 100
1 2
2 1
1 2

Output:

Case 1:
600
630

Note

Dataset is huge. Use faster I/O methods.

Problem Setter: Ahmad Faiyaz

Special Thanks: Syed Shahriar Manjur