

Problem J – Job Allocator

The Infrastructure Consortium for Public Compute (ICPC) is a network of computers ran by volunteers from all around the world that share compute resources with one another. Contributors are able to plug and unplug their machines on the network, and also to run compute jobs on the network machines. With the ICPC, important projects that would otherwise have prohibitive infrastructure costs (like running online judges for programming competitions) become viable endeavors.

As great as it sounds on paper, for now, the ICPC is just a dream. To make it work, a key piece of software is missing: the job allocator. This is where you come in: the community is counting on you to make this important (but voluntary, of course) contribution.

The network is extremely dynamic: machines connect and disconnect all the time. The job allocator needs to keep track of the machines that are currently connected and which resources they share. There are several types of resources, such as CPU cores, GPUs, and SSD disks. One machine can share one or more resources, possibly more than one of the same type. Moreover, at any point in time, users can request machines to run compute jobs. For that, they specify a list of resources that a machine needs to run their job, and the job allocator has to determine how many of the currently connected machines have all the required resources to run the job. For example, for a job that needs one CPU core and two GPUs, the allocator would need to count how many machines have at least one CPU core, and two or more GPUs.

Your task is to simply count how many connected machines satisfy each job’s resource requirements, since another volunteer took on the task of implementing the actual assignment of jobs to machines. The whole ICPC community depends on you. Are you able to help?

Input

The first line contains two integers N ($1 \leq N \leq 10^5$) and K ($1 \leq K \leq 8$), indicating respectively the number of network events that must be processed and the number of types of resources that are available on the ICPC. Events are described in chronological order in the next N lines, one event per line. There are three types of events.

If the event represents that a new machine is being connected to the network, the line contains the uppercase letter “C”, followed by an integer R ($1 \leq R \leq 8$) indicating the number of resources the machine is sharing, followed by R integers T_1, T_2, \dots, T_R ($1 \leq T_i \leq K$ for $i = 1, 2, \dots, R$), describing the type of each of the shared resources. New machines are implicitly assigned unique sequential integer identifiers by the ICPC, starting at 1.

When the event represents that a machine is being disconnected from the network, the line contains the uppercase letter “D”, followed by an integer indicating the identifier of the machine. It is guaranteed that this identifier corresponds to a valid connected machine.

Finally, if the event represents that a user wants to run a job, the line contains the uppercase letter “J”, followed by an integer R ($1 \leq R \leq 8$) indicating the number of resources the job needs, followed by R integers T_1, T_2, \dots, T_R ($1 \leq T_i \leq K$ for $i = 1, 2, \dots, R$), describing the type of each of the required resources. It is guaranteed that the input contains at least one event of this type.

Output

Output a line for each event of type “J”. The line must contain an integer indicating the number of machines that at the moment of the event are connected to the network and provide all the requested resources. Write the results in chronological order, that is, using the same order as the input.

ICPC Latin American Regional – 2020

Sample input 1 3 2 C 3 1 1 2 J 2 1 2 J 2 2 2	Sample output 1 1 0
Sample input 2 11 3 C 2 1 2 J 1 3 C 3 1 2 3 J 1 3 J 1 1 D 2 J 1 3 J 2 1 2 J 2 1 1 D 1 J 1 2	Sample output 2 0 1 2 0 1 0 0 0 1 0 0