

Fraud Checker

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

points: 100 · problem code: fraudchecker · author: glipR

You are helping a bank spot some fraudulent transactions. You've got some n people, who have made m transactions. Each of these people have a balance, and for some reason, people are going into negative balance as a result of some of these transactions.

It turns out there is a fraudulent person in this register, and all transactions involving this person is fraudulent, and never actually occurred. If this fraudulent person was removed, then the remaining transactions would make sense.

It's your job to find this fraudulent person and report them. Note that there will always be exactly one person who fits this description.

Input

Input will start with two space separated integers, n and m , the number of people and number of transactions. The next n lines contain two values, the name of a particular person and their starting balance. Names are of length ≤ 20 and only contain lowercase ascii characters. The next m lines contain transactions. Each line contains a string of the form `<name1> gives <name> <amount>`. All amounts are integers.

Output

Output a single string - the name of a person that, if removed, all remaining transactions would never have the balances dip below ~ 0 .

Example

Example Input

```
5 6
alice 20
bob 10
clarissa 15
daniel 50
erin 0
daniel gives erin 20
erin gives bob 20
daniel gives bob 50
clarissa gives alice 15
clarissa gives erin 5
bob gives alice 15
```

Example Output

```
erin
```

As is, multiple people fall below 0 balance throughout the transactions. But, if erin is removed from the equation, we have the following transactions:

```
daniel gives 50 bob
clarissa gives alice 15
bob gives alice 15
```

Which works.

Constraints

- $1 \leq n \leq 50$
- $1 \leq m \leq 500$
- All values are less than or equal to 10^7 .