

---

# Compressed LCS

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            3 seconds  
Memory limit:         512 megabytes

Bobo has two integer sequences  $A$  and  $B$ , both in compressed form.  $A = c_1^{a_1} c_2^{a_2} \dots c_n^{a_n}$  means that  $A$  begins with  $a_1$  copies of the integer  $c_1$ , followed by  $a_2$  copies of the integer  $c_2$ ,  $a_3$  copies of the integer  $c_3$ , and so on.  $B = d_1^{b_1} d_2^{b_2} \dots d_m^{b_m}$  is of similar format.

Bobo would like to find the LCS (longest common subsequence) for  $A$  and  $B$ . Recall that sequence  $C$  is a subsequence of  $A$  if and only if  $C$  can be obtained by deleting some (maybe all, maybe none) elements from  $A$ .

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains two integers  $n$  and  $m$  ( $1 \leq n, m \leq 2000$ ).

The  $i$ -th of the following  $n$  lines contains two integers  $c_i$  and  $a_i$ . And the  $i$ -th of the last  $m$  lines contains two integers  $d_i$  and  $b_i$ . The constraints are:  $1 \leq a_i, b_i, c_i, d_i, \sum_{i=1}^n a_i, \sum_{i=1}^m b_i \leq 10^9$ ,  $c_i \neq c_{i-1}$ ,  $d_i \neq d_{i-1}$ .

It is guaranteed that the sum of  $n$  and the sum of  $m$  both do not exceed 2000.

## Output

For each test case, output an integer which denotes the length of the LCS.

## Example

standard input	standard output
1 3	2
1 2	3
1 1	999
2 1	
1 2	
4 4	
1 1	
2 1	
3 1	
4 1	
1 1	
3 1	
2 1	
4 1	
1 1	
1000000000 999	
1000000000 1000	