

The 41st Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition



Problem H

Project Team

Time Limit: 1 Second

A team composed of m members is going to carry out a project in n contiguous working days. You are helping the team to set up the vacation schedules of the members. Every member is bound to work for at least p and at most p' working days for the project and, in particular, the number of team members required to work on the i -th working day is between q_i and q_i' , inclusive, for $i = 1, \dots, n$. Also, each member has a series of vacation plans

$$(d_1, [r_1, r'_1]), (d_2, [r_2, r'_2]), \dots, (d_k, [r_k, r'_k]),$$

which indicate that the member wants to take at least d_i vacation days from the working-day period $[r_i, r'_i]$ for each $i = 1, \dots, k$, and wants to work on a day not included in the union of the period $[r_i, r'_i]$ over all i , i.e., $\bigcup_{i=1}^k \{r: r_i \leq r \leq r'_i\}$. Note that the vacation days are presumably not fixed in a vacation plan. If a member has a vacation plan $(2, [7, 9])$ for instance, he/she may take a two-day vacation $\{7, 8\}$, $\{7, 9\}$, $\{8, 9\}$, or happily a three-day vacation $\{7, 8, 9\}$; whereas for some vacation plans, say $(2, [3, 4])$, the vacation days are fixed.

Given the information on the project team and the vacation plans for the team members, you are to write a computer program that determines if it is possible to assign vacation days to each of the members in a way that everyone is happy, subject to the aforementioned constraints. Such an assignment of vacation days is called a vacation schedule. It is assumed that the members are indexed from 1 to m .

For example, suppose you are given $m = 3$, $n = 5$, and $(p, p') = (2, 3)$, as well as (q_i, q_i') for working day $i \in \{1, \dots, 5\}$ shown in the table below (left) and the vacation plans for member $j \in \{1, 2, 3\}$ shown in the table (center). You can make vacation schedules of the members that satisfy all the constraints, as shown in the table below (right).

i -th day	(q_i, q_i')
1	(2, 2)
2	(2, 3)
3	(1, 2)
4	(1, 3)
5	(1, 2)

member j	vacation plans
1	$(2, [1, 3])$
2	$(2, [2, 3]), (1, [4, 5])$
3	$(2, [3, 5])$

member j	vacation days
1	1, 3
2	2, 3, 5
3	4, 5

Input

Your program is to read from standard input. The first line of the input contains four positive integers m , n , p and p' whose meaning has been described above, where $m \leq 100$, $n \leq 100$, and $p \leq p' \leq n$. In the following n lines, each line contains two positive integers q_i and q_i' for $i = 1$ to n , where $q_i \leq q_i' \leq m$ for all i . Then, m lines follow, where the j -th line contains the vacation plans for the member j . The vacation plans of a member, denoted by $(d_1, [r_1, r'_1]), (d_2, [r_2, r'_2]), \dots, (d_k, [r_k, r'_k])$, is represented by a sequence of $3k + 1$ positive integers as follows: $k, d_1, r_1, r'_1, d_2, r_2, r'_2, \dots, d_k, r_k, r'_k$. The number of vacation plans of a member is no more than 20. You may assume that $r_i \leq r'_i \leq n$ and $d_i \leq r'_i - r_i + 1$ for all $i \in \{1, \dots, k\}$, $r_1 < r_2 < \dots < r_k$, and moreover two periods $[r_a, r'_a]$ and $[r_b, r'_b]$ are disjoint whenever $a \neq b$, i.e., they do not contain a common working day if $a \neq b$ and $a, b \in \{1, \dots, k\}$.

Output

Your program is to write to standard output. The first line must contain an integer indicating whether or not there exist vacation schedules of the team members that satisfy all the constraints. If yes, the integer must be 1; otherwise -1. When and only when the first line is 1, it must follow m lines containing, one by one, the vacation schedules for member $j = 1$ to m , where the vacation schedule of a member is described by the number of vacation days followed by the vacation days in ascending order.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
3 5 2 3 2 2 2 3 1 2 1 3 1 2 1 2 1 3 2 2 2 3 1 4 5 1 2 3 5	1 2 1 3 3 2 3 5 2 4 5
Sample Input 2	Output for the Sample Input 2
3 5 3 4 2 3 2 3 2 3 2 3 2 3 1 1 2 2 1 1 3 3 1 1 2 3	-1