B. Zayin and Elements

Zayin has n elements (numbered from 1 to n) and m props (numbered from 1 to m). Each prop can remove some specific elements. Prop i has shield a_i, base HP b_i, and bonus HP c_i. When Zayin wants to remove an element, he should obey the following rules:

Zayin should choose an alive prop (e.g. prop i) which can be used to remove this element. (Prop i is alive when the sum of its shield point, base HP and bonus HP is strictly greater than 0. That is to say, $a_i + b_i + c_i > 0$.)

1. If the shield (a_i) is greater than 0, the shield will be subtracted by 1, then go to 4, else go to 2.
2. If the base HP (b_i) is greater than 0, the base HP will be subtracted by 0.5, then go to 4, else go to 3.
3. The bonus HP (c_i) will be subtracted by 1, then go to 4.
4. Remove the element, and the process is finished.

Now Zayin wants to remove all the elements and maximized $\sum_{i=1}^{m} \lfloor b_i + c_i \rfloor$ ($\lfloor x \rfloor$ means the floor of x, the biggest integer which is not greater than x), but he is too busy so it is time for you to solve this problem.

Input

The first line of input contains an integer $T(1 \le T \le 5)$, denoting the number of test cases.

For each test case, the first line contains an integer $n(1 \le n \le 100)$, the number of elements.

The second line contains an integer $m(1 \le m \le 20)$, the number of props.

In the next m lines, the ith line firstly comes three integers a_i, b_i, c_i $(a_i, b_i, c_i \ge 0,\ a_i + b_i + c_i \le 10)$, representing the shield, base HP and bonus HP of prop i. Then follows a number t_i, indicates the number of elements can be removed by this prop. After that comes t_i numbers, representing the element numbers.

Output

For each test case, output the maximal $\sum_{i=1}^{m} \lfloor b_i + c_i \rfloor$. (or "$-1$" if there is no way to remove all the elements)

Sample

Input	Output
2 5 2 2 3 1 2 3 1 1 1 1 3 4 5 2 5 2 2 3 1 1 3 1 0 1 2 1 2	5 -1