

In a large church in New Zealand, after the church service, the priest gives an empty collection box that can hold c dollar coins to the “light-fingered usher”. The usher then passes the collection box to a nearby parishioner. When receiving the box a parishioner adds a few dollar coins, then passes it to another nearby parishioner. When the light-fingered usher receives the box, he quietly removes one dollar coin from the box, slips it into his pocket, and passes the box to a nearby parishioner.

The behavior of the usher is given by a set of parishioners to whom he may pass the box. The behavior of a parishioner is given by a set of rules, each consisting of a donation of at least two dollar coins, and another parishioner to whom the box is passed after he places the coins in the box. As soon as the box is full, containing c coins, it is immediately passed to the priest, even when a parishioner cannot finish entering all the coins specified by the chosen rule.

Your problem is to compute the maximal possible gain for the usher, which is achieved when the parishioners always choose a rule that leads to the biggest number of coins in the usher’s pocket.

Input

The input consists of a line containing the number c of datasets, followed by c datasets, followed by a line containing the number ‘0’.

The first line of each dataset contains two numbers b and p , separated by a blank. The number b specifies the capacity of the box, and p the number of parishioners. You can assume that $1 \leq b \leq 1000000$ and $1 \leq p \leq 500$. The next line specifies the behavior of the usher, represented by integers separated by a blank. The first integer specifies the number q of parishioners, to which the usher may pass the box. The next q integers each represent a parishioner to whom he may pass the box. The parishioners are numbered from 1 to p . Each line i ($1 \leq i \leq p$) of the next p lines of each dataset describes the behavior of parishioner i , and consists of integers separated by a blank. The first integer of the line specifies the number of rules for the parishioner. Each rule is represented by a pair of integers, the first of which specifies the number of coins to give, which must be equal or greater than 2. The second number indicates the parishioner to receive the box next, where the number 0 identifies the usher. When there are two or more rules, the parishioner may choose to apply any of them. You can assume that the number of rules per parishioner is at least 1 and less than or equal to 1000; there may be multiple rules with the same parishioner to receive the box.

Output

The output consists of one line for each dataset. The c -th line contains the maximal number of coins that the usher can obtain for dataset c .

Note: The dataset below specifies that the box can hold up to 10 coins, and that there are two parishioners. The usher may pass the box to either one of the two parishioners, as indicated by the second line. Parishioner 1 has two rules, and can choose either one, when the box is passed to him. The first rule says to put down 6 dollar coins and pass the box to the usher (indicated by ‘0’), and the second rule is put down 4 dollar coins and pass the box to parishioner 2. The last line specifies one single rule for parishioner 2, who must put down 5 dollar coins, and then pass the box to the usher.

The usher can obtain the maximal amount of 2 dollars by passing the box to parishioner 2, who passes it back to the usher. At that point, there are 5 dollars on the box. After removing one dollar, the box goes with 4 dollars to parishioner 2, and back to the usher, now with 8 dollars. The usher removes another dollar, and the box goes to parishioners 2 with 7 dollars. Parishioner 2 starts to apply his rule, and manages to put three more coins into the box, after which the box is full and goes to the priest. The usher ends up with two dollar coins in his pocket.

Sample Input

```
1
10 2
2 1 2
2 6 0 4 2
1 5 0
0
```

Sample Output

```
2
```