

7545 Calculating Taxes

The Jones family is a very wealthy family who has lived in an area of the country for many generations. In this area of the country, houses are scattered around with dirt roads connecting them. Two houses are called *neighbours* if there is a dirt road connecting them. Using the dirt roads, there is exactly one way to get from any house to any other house without walking on some road at least twice. There is a new king in the country and he has decided that he will be taking taxes in a rather weird way.

Each household is to pick some divisor of their income from the last year (this value could be their full income if they wish). If any neighbours pick a number whose greatest common divisor is larger than 1, then the king will take every dollar from every person in the country. On the other hand, if each pair of neighbours pick numbers whose greatest common divisor is 1, then each household may keep whatever number they chose.

Obviously, one option is for each household to take \$1 each, but that is not very profitable. Can you help them maximize the total amount of money that they can keep?

Input

The input file contains several test cases, each of them as described below.

The first line will contain one integer n ($2 \leq n \leq 250$) denoting the number of houses. The next n lines contain the information about each household. The first line will contain information about house 1, the second line will contain information about house 2, etc.

Each of these lines will begin with two integers, I_k ($1 \leq I_k \leq 200\,000\,000$) and m_k ($1 \leq m_k \leq n-1$), denoting the income of the k -th household and the number of neighbours of the k -th household. Then follow m_k distinct integers denoting the neighbours of the k -th household (each will be between 1 and n , inclusive, and none of these will be k).

Each road is bidirectional and will be listed twice in the input (once for each endpoint of the road).

Output

For each test case, output one integer, the maximum total amount of money that the households can keep.

Sample Input

```
4
2 1 2
3 2 1 3
2 2 2 4
3 1 3
4
3 1 2
9 2 1 3
3 2 2 4
9 1 3
6
4 3 3 4 6
3 1 3
18 3 1 2 5
```

6 1 1
9 1 3
10 1 1

Sample Output

10
20
37