**PIGS**

|  |  |  |
| --- | --- | --- |
| **Time Limit:** 1000MS |  | **Memory Limit:** 10000K |
|  |  |  |

**Description**

Mirko works on a pig farm that consists of M locked pig-houses and Mirko can't unlock any pighouse because he doesn't have the keys. Customers come to the farm one after another. Each of them has keys to some pig-houses and wants to buy a certain number of pigs.   
All data concerning customers planning to visit the farm on that particular day are available to Mirko early in the morning so that he can make a sales-plan in order to maximize the number of pigs sold.   
More precisely, the procedure is as following: the customer arives, opens all pig-houses to which he has the key, Mirko sells a certain number of pigs from all the unlocked pig-houses to him, and, if Mirko wants, he can redistribute the remaining pigs across the unlocked pig-houses.   
An unlimited number of pigs can be placed in every pig-house.   
Write a program that will find the maximum number of pigs that he can sell on that day.

**Input**

The first line of input contains two integers M and N, 1 <= M <= 1000, 1 <= N <= 100, number of pighouses and number of customers. Pig houses are numbered from 1 to M and customers are numbered from 1 to N.   
The next line contains M integeres, for each pig-house initial number of pigs. The number of pigs in each pig-house is greater or equal to 0 and less or equal to 1000.   
The next N lines contains records about the customers in the following form ( record about the i-th customer is written in the (i+2)-th line):   
A K1 K2 ... KA B It means that this customer has key to the pig-houses marked with the numbers K1, K2, ..., KA (sorted nondecreasingly ) and that he wants to buy B pigs. Numbers A and B can be equal to 0.

**Output**

The first and only line of the output should contain the number of sold pigs.

**Sample Input**

3 3

3 1 10

2 1 2 2

2 1 3 3

1 2 6

**Sample Output**

7