Problem A. Hardwood Species

Input:	`Standard`
Output:	`Standard`
Judge:	`UVa`

Hardwoods are the botanical group of trees that have broad leaves, produce a fruit or nut, and generally go dormant in the winter.

America's temperate climates produce forests with hundreds of hardwood species – trees that share certain biological characteristics. Although oak, maple and cherry all are types of hardwood trees, for example, they are different species. Together, all the hardwood species represent 40 percent of the trees in the United States.

On the other hand, softwoods, or conifers, from the Latin word meaning "cone-bearing," have needles. Widely available US softwoods include cedar, fir, hemlock, pine, redwood, spruce and cypress. In a home, the softwoods are used primarily as structural lumber such as 2x4s and 2x6s, with some limited decorative applications.

Using satellite imaging technology, the Department of Natural Resources has compiled an inventory of every tree standing on a particular day. You are to compute the total fraction of the tree population represented by each species.

The first line is the number of test cases, followed by a blank line. Each test case of your program consists of a list of the species of every tree observed by the satellite; one tree per line. No species name exceeds 30 characters. There are no more than 10,000 species and no more than 1,000,000 trees. There is a blank line between each consecutive test case.

For each test case print the name of each species represented in the population, in alphabetical order, followed by the percentage of the population it represents, to 4 decimal places. Print a blank line between 2 consecutive data sets.

Samples

Input	Output
1	Ash 13.7931
Red Alder	Aspen 3.4483
Ash	Basswood 3.4483
Aspen	Beech 3.4483
Basswood	Black Walnut 3.4483
Ash	Cherry 3.4483
Beech	Cottonwood 3.4483
Yellow Birch	Cypress 3.4483
Ash	Gum 3.4483
Cherry	Hackberry 3.4483
Cottonwood	Hard Maple 3.4483
Ash	Hickory 3.4483
Cypress	Pecan 3.4483
Red Elm	Poplan 3.4483
Gum	Red Alder 3.4483
Hackberry	Red Elm 3.4483
White Oak	Red Oak 6.8966
Hickory	Sassafras 3.4483
Pecan	Soft Maple 3.4483
Hard Maple	Sycamore 3.4483
White Oak	White Oak 10.3448
Soft Maple	Willow 3.4483
Red Oak	Yellow Birch 3.4483
Red Oak	
White Oak	
Poplan	
Sassafras	
Sycamore	
Black Walnut	
Willow	

URL:
http://uva.onlinejudge.org/index.php?
option=onlinejudge&page=show_problem&problem=1167

Problem B. Dominos

Input:	`Standard`
Output:	`Standard`
Judge:	`UVa`

Dominos are lots of fun. Children like to stand the tiles on their side in long lines. When one domino falls, it knocks down the next one, which knocks down the one after that, all the way down the line. However, sometimes a domino fails to knock the next one down. In that case, we have to knock it down by hand to get the dominos falling again.

Your task is to determine, given the layout of some domino tiles, the minimum number of dominos that must be knocked down by hand in order for all of the dominos to fall.

Input

The first line of input contains one integer specifying the number of test cases to follow. Each test case begins with a line containing two integers, each no larger than 100 000. The first integer n is the number of domino tiles and the second integer m is the number of lines to follow in the test case. The domino tiles are numbered from 1 to n. Each of the following lines contains two integers x and y indicating that if domino number x falls, it will cause domino number y to fall as well.

Output

For each test case, output a line containing one integer, the minimum number of dominos that must be knocked over by hand in order for all the dominos to fall.

Samples

Input	Output

1
3 2
1 2
2 3
	1

URL:
http://uva.onlinejudge.org/index.php?
option=onlinejudge&page=show_problem&problem=2499

Problem C. Internet Bandwidth

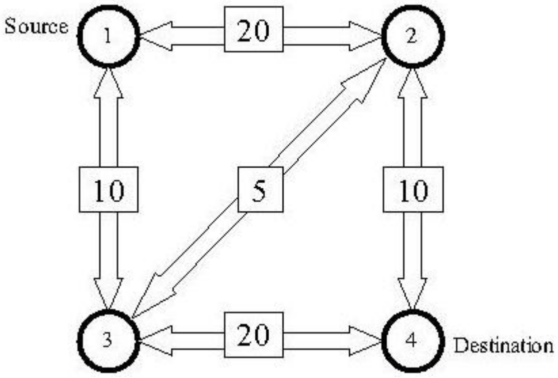
Input:	`Standard`
Output:	`Standard`
Judge:	`UVa`

On the Internet, machines (nodes) are richly interconnected, and many paths may exist between a given pair of nodes. The total message-carrying capacity (bandwidth) between two given nodes is the maximal amount of data per unit time that can be transmitted from one node to the other. Using a technique called packet switching, this data can be transmitted along several paths at the same time.

For example, the following figure shows a network with four nodes (shown as circles), with a total of five connections among them. Every connection is labeled with a bandwidth that represents its data-carrying capacity per unit time.

In our example, the bandwidth between node 1 and node 4 is 25, which might be thought of as the sum of the bandwidths 10 along the path 1-2-4, 10 along the path 1-3-4, and 5 along the path 1-2-3-4. No other combination of paths between nodes 1 and 4 provides a larger bandwidth.

You must write a program that computes the bandwidth between two given nodes in a network, given the individual bandwidths of all the connections in the network. In this problem, assume that the bandwidth of a connection is always the same in both directions (which is not necessarily true in the real world).



Input

The input file contains descriptions of several networks. Every description starts with a line containing a single integer n ($2 \leq n \leq 100$), which is the number of nodes in the network. The nodes are numbered from 1 to n. The next line contains three numbers s, t, and c. The numbers s and t are the source and destination nodes, and the number c is the total number of connections in the network. Following this are c lines describing the connections. Each of these lines contains three integers: the first two are the numbers of the connected nodes, and the third number is the bandwidth of the connection. The bandwidth is a non-negative number not greater than 1000.

There might be more than one connection between a pair of nodes, but a node cannot be connected to itself. All connections are bi-directional, i.e. data can be transmitted in both directions along a connection, but the sum of the amount of data transmitted in both directions must be less than the bandwidth.

A line containing the number 0 follows the last network description, and terminates the input.

Output

For each network description, first print the number of the network. Then print the total bandwidth between the source node s and the destination node t, following the format of the sample output. Print a blank line after each test case.

Samples

Input	Output

4
1 4 5
1 2 20
1 3 10
2 3 5
2 4 10
3 4 20
0
	Network 1
	The bandwidth is 25.

URL:
http://uva.onlinejudge.org/index.php?
option=com_onlinejudge&Itemid=8&page=show_problem&problem=761

Problem D. Fire Station

Input:	`Standard`
Output:	`Standard`
Judge:	`UVa`

A city is served by a number of fire stations. Some residents have complained that the distance from their houses to the nearest station is too far, so a new station is to be built. You are to choose the location of the fire station so as to reduce the distance to the nearest station from the houses of the disgruntled residents.

The city has up to 500 intersections, connected by road segments of various lengths. No more than 20 road segments intersect at a given intersection. The location of houses and firestations alike are considered to be at intersections (the travel distance from the intersection to the actual building can be discounted). Furthermore, we assume that there is at least one house associated with every intersection. There may be more than one firestation per intersection.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The first line of input contains two positive integers: f,the number of existing fire stations ($f \leq 100$) and i, the number of intersections ($i \leq 500$). The intersections are numbered from 1 to i consecutively. f lines follow; each contains the intersection number at which an existing fire station is found. A number of lines follow, each containing three positive integers: the number of an intersection, the number of a different intersection, and the length of the road segment connecting the intersections. All road segments are two-way (at least as far as fire engines are concerned), and there will exist a route between any pair of intersections.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

You are to output a single integer: the lowest intersection number at which a new fire station should be built so as to minimize the maximum distance from any intersection to the nearest fire station.

Samples

Input	Output

1
1 6
2
1 2 10
2 3 10
3 4 10
4 5 10
5 6 10
6 1 10
	5

URL:
http://uva.onlinejudge.org/index.php?
option=com_onlinejudge&Itemid=8&page=show_problem&problem=761

Problem E. Cut Ribbon

Input:	`Standard`
Output:	`Standard`
Judge:	`CodeForces`

Polycarpus has a ribbon, its length is n. He wants to cut the ribbon in a way that fulfils the following two conditions:

- After the cutting each ribbon piece should have length a, b or c.

- After the cutting the number of ribbon pieces should be maximum.

Help Polycarpus and find the number of ribbon pieces after the required cutting.

Input

The first line contains four space-separated integers n, a, b and c ($1 \leq n, a, b, c \leq 4000$) - the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly. The numbers a, b and c can coincide.

Output

Print a single number - the maximum possible number of ribbon pieces. It is guaranteed that at least one correct ribbon cutting exists.

Samples

Input	Output
`5 5 3 2`	`2`
`7 5 5 2`	`2`

Note: In the first example Polycarpus can cut the ribbon in such way: the first piece has length 2, the second piece has length 3.

In the second example Polycarpus can cut the ribbon in such way: the first piece has length 5, the second piece has length 2.

URL:
http://codeforces.com/problemset/problem/189/A

Credits

LaTeX edition by:

Daniel Cañizares Corrales.
Professor, Universidad Católica de Oriente.

All problems can be found on the provided URLs.
This compilation was made for educational purposes.