

2014 Taiwan National Collegiate Programming Contest

Preliminary Round: Oct. 4. 2014

- Problems: There are 8 problems (18 pages in all) in this packet.
- Problem Input: Input to the problems are through standard input. Each input file may contain one or more test cases. Test cases may be separated by any delimiter as specified in the problem statements.
- Problem Output: All output should be directed to the standard output (screen output).
- Time Limit: The judges will run each submitted program with certain time limit (given in the table below).

Table 1: Problem Information

	Problem Name	Time Limit
Problem A	What is the largest value?	1 sec.
Problem B	Finding Odious Numbers	1 sec.
Problem C	A=LU Decomposition	3 sec.
Problem D	Sink Selection	1 sec.
Problem E	Search the Divergent	1 sec.
Problem F	Battery of Electric Cars	3 sec.
Problem G	Secret Clubs	3 sec.
Problem H	Water Tanks and Hoses	5 sec.

Problem A

What is the largest value?

Time Limit: *1 Second*

You are given two 2-dimensional arrays A and B . A has n rows and m columns. B has m rows and p columns. Let $C = A \cdot B$ be the resulting array by multiplying A and B . Thus C has n rows and p columns. Your job is to find out the largest value in the entries of C .

Technical Specification

1. Each entry in A and B is an integer in the range of $-1,000$ to $1,000$.
2. $0 < n, m, p \leq 100$.

Input File Format

The first line of the input file contains an integer, which is the number of test cases to follow. For each test case, the first line are 3 integers indicating n , m and p . The next n lines contain the n rows of A in this test case. In each line, the m entries in this row are listed as integers separated by a blank. Then the next m lines contain the m rows of B in this test case. In each line, the p entries in this row are listed as integers separated by a blank.

Output Format

For each test case, the largest value in the entries of C is in a line.

Sample Input

```
2
2 3 2
1 2 3
-1 3 0
4 0
2 2
-1 9
2 2 2
-1000 1000
3 0
2 9
7 8
```

Output for the Sample Input

```
31
5000
```


Problem B

Finding Odious Numbers

Time Limit: *1 Second*

An *odious number* is a nonnegative integer that has an odd number of 1s in its binary expansion. The first few odious numbers are therefore 1, 2, 4, 7, 8, 11, 13, 14, 16, 19, \dots .

Technical Specification

Note that the value of n which will be tested is less than or equal to 32767.

Input File Format

The first line of the input file contains an integer denoting the number of test cases. There are at most 10 test cases. In each test case, there is a line containing a nonnegative integer n .

Output Format

For each test case, output 1 in a line if it is an odious number; otherwise, output 0 in a line.

Sample Input

```
10
7
20
15
8
16
9
13
12
11
22
```

Output for the Sample Input

```
1
0
0
1
1
0
1
0
1
1
```


Problem C

A=LU Decomposition

Time Limit: 3 seconds

Many problems of science and engineering can be reduced to solving a linear system of equations. In mathematics, given a matrix $A \in R^{n \times n}$ and a column vector $\mathbf{b} \in R^n$, one desires to design an efficient program to compute a precise \mathbf{x} for $A\mathbf{x} = \mathbf{b}$. One of the fundamental techniques for solving the aforementioned problem is to decompose the matrix A into the product of two simpler matrices L and U , where L is unit lower triangular (the diagonal elements are all 1s and upper diagonal elements are all 0s) and U is upper triangular (the lower diagonal elements are all 0s). For example,

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 2 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = LU$$

This problem asks you to write a program to decompose a given $n \times n$ matrix A into the product of L and U with the results stored in the same input matrix A . *An algorithm is listed for your reference.*

Algorithm for $A = LU$.

```
for  $i = 1, 2, \dots, n - 1$ 
  for  $k = i + 1, i + 2, \dots, n$ 
     $r \leftarrow A(k, i) / A(i, i)$  if  $A(i, i) \neq 0$ ;
    for  $j = i + 1, i + 2, \dots, n$ 
       $A(k, j) \leftarrow A(k, j) - r * A(i, j)$ ;
    endfor
     $A(k, i) = r$ ;
  endfor
endfor
```

Note that for each matrix A given in this contest, $A(i, i)$ will never be near zero, so that $A = LU$ is always feasible.

Input File Format

The first line of the input file contains one integer, $K \leq 5$, indicating the number of test cases to come. Each test case begins with a line containing a positive integer h indicating the dimension of matrix A followed by the contents of A in h lines, each line has h real numbers separated by spaces. In each of the test case, the dimension h is restricted as $2 \leq h \leq 5$, and each element of the matrix is in $(-32, 32)$.

Output Format

The first line of the output file is the number of test cases K . The remaining has the same format as the input file, except that the contents of the matrix A are replaced by the corresponding output results. That is, the diagonal and upper triangular parts are the results of matrix U and the lower triangular parts are the results of matrix L . The precision is rounded up to the **first place** after the decimal point.

Sample Input

```
3
2
2.0 1.0
0.0 -3.0
3
3.0 1.0 1.0
6.0 4.0 3.0
-3.0 5.0 3.0
4
2.0 2.0 2.0 2.0
0.0 3.0 3.0 3.0
-2.0 -2.0 0.0 0.0
0.0 -3.0 -3.0 0.0
```

Sample Output

```
3
2
2.0 1.0
0.0 -3.0
3
3.0 1.0 1.0
2.0 2.0 1.0
-1.0 3.0 1.0
4
2.0 2.0 2.0 2.0
0.0 3.0 3.0 3.0
-1.0 0.0 2.0 2.0
0.0 -1.0 0.0 3.0
```

Problem D

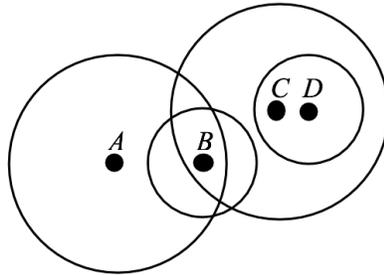
Sink Selection

Time Limit: 1 Second

Wireless sensor networks are networks comprising a large number of small size and low cost sensing devices, which are deployed onto some area of interest through any viable approach. Every sensor may be designed to detect various kinds of signals (e.g., temperatures, humidity, luminosity, or voice) and then relays the collected information to the users by propagating hop-by-hop through intermediate sensing nodes within its communication radius.

Sensor networks usually connect to the outside world through the so called sinks (may be regarded as gateways). All data collected by sensor nodes are sent to the sink, and then the sink relays such data to remote users or servers through the Internet, satellite, or any viable medium. One fundamental issue in wireless sensor networks is selecting a specific sensor among all sensors (in a plane) as the sink so that every other sensor has a communication path to the sink.

Suppose that the communication range (radius) of sensor A is R_A unit length. Sensor A can send packets to sensor B directly, when the distance between A and B is less than or equal to R_A . Specifically, sensor A can send packets to sensor B directly if $(x_1 - x_2)^2 + (y_1 - y_2)^2 \leq R_A^2$, where (x_1, y_1) and (x_2, y_2) are, respectively, the locations of A and B . Note that when sensor A can send packets to sensor B directly, sensor B maybe cannot do that conversely when the communication ranges of these sensors are different. For simplicity, we assume that the communication area of each sensor is modeled by a circle and the location of the sensor is marked by a black dot in the center of a circle (see below).



In the above figure, there are four sensors A, B, C, D deployed in the plane. Suppose that the radiuses of sensor A and C are 4 unit lengths (i.e. $R_A = R_C = 4$), and the radiuses of sensor B and D are 2 unit length (i.e. $R_B = R_D = 2$). Since the circle of A (C) covers the black dot of B in the figure, A (C) can send packets to B directly. However, B cannot send packets to A (C) directly, because the circle of B cannot cover the black dot of A (C). Similarly, D can send packets to C directly, but D cannot send packets to B or A directly. Obviously, only sensor B can be selected as the sink in the above figure.

Please write a program to count how many sensors can be selected as the sink for a given sensor network.

Input File Format

There are at most 10 test cases. The first line of each instance consists of an integer N ($1 \leq N \leq 100$), where N is the number of sensors in the network. Each of the following N lines consists of three integers x, y, R (separated by a space) ($-9 \leq x \leq 99, -99 \leq y \leq 99, 1 \leq R \leq 99$), which indicate the locations (x-coordinate and y-coordinate) of a sensor in the plane and the radius R of the sensor. No two sensors have the same location in the same instance. The last test case will be followed by a line containing $N = 0$.

Output Format

The output for each instance should contain an integer denoting the number of sensors can selected as the sink in the given network.

Sample Input

```
4
1 6 4
4 6 2
7 8 4
8 8 2
4
1 6 4
4 6 4
7 8 4
8 8 4
0
```

Output for the Sample Input

```
1
4
```

Problem E

Search the Divergent

Time Limit: 1 Second

In year 2500, after a civil war which killed most of people, survivors in the world strive to build a long, lasting, and stable society. In order to assure the stability of society, society is divided into five factions: A, B, C, D and E . The world is built by the guidance from a group of geneticists. In principle, if humans are genetically engineered into the five kinds of people, these people shall form a peaceful, harmonious, and stable society and wars can be forever prevented.

Unfortunately, genetically engineering humans at the level of human virtues is not always successful. So, the government lets the 16-year olds choose one faction from A, B, C, D, E in an open Choosing Ceremony. After the Choosing Ceremony, they leave their homes and begin a new life to live with the humans of the same faction.

On the other hand, before the Choosing Ceremony, a youngster may approach elders to establish the mentor/apprentice relationship to explore themselves. The elders are older than the youngster but they are not necessarily older than 16-year old. So, an elder may not choose his faction yet. Anyone can be a mentor regardless of his age, as long as he is older than his apprentices. To the government, if a youngster is genetically engineered correctly, he will choose elders of a same faction as his mentors without knowing why.

After years of genetic experiments, a group of humans called “the divergent” are identified. These humans are considered as failed products of genetic engineering. Although the divergents will choose one faction in the Choosing Ceremony, they do not really behave the same as people in that faction. Studies show that they can endanger the stability of society.

The government begins to search these divergents and exile them into a desolation area. After that, the divergents begin to hide themselves as much as they can. Now, the government wants to execute a big-data analysis project to search the divergents. In the past, the mentor/apprentice relationships are saved by the local governments.

To a non-divergent youngster, genetic instinct would trigger them to find mentors who behave like his designated faction. For example, a non-divergent youngster X , who is genetically engineered to be faction A , would find mentors who behave like faction A . However, a mentor of X can be a divergent and thus can fake his behaviours for being like faction A . A divergent, however, is not genetically engineered successfully, so he/she tends to try different faction’s living ways. As a result, a divergent tends to find mentors from different factions to satisfy their curiosity.

Given all the relationships, you, as part of the evil government, are ordered to write an analysis program to find the divergents. You know this is impossible. However, better be safe than sorry, you propose to find “potential” divergents. A potential divergent is who has mentors (or mentor’s mentors, etc.) that belong to different factions. Another way of becoming a potential divergent is who chooses his/her faction at the Choosing Ceremony but his/her choice is different from his/her mentors and his/her mentors are of a same faction.

Input File Format

The test data is a series of pair $(x \ y)$. x is the identification number of a citizen which ranges from $(1..999999999)$. If y is an identification number, then this pair describes that x is an apprentice of y . An apprentice can have maximum 10 mentors but a mentor can have unlimited apprentices if he/she likes. If y is not an integer number then it must be

of 'A','B','C','D', or 'E', which represents an event that x chooses his/her faction in the Choosing Ceremony. When the data is processed, at any time, "@find" can be used to ask your program to output potential divergents. Please output the divergent's identification number by ascending order. If there is no divergent, please output "none". When "@end" is read, it is the end of test data.

Sample Input

```
1 B
2 B
4 C
5 C
6 1
6 2
7 2
7 3
8 3
8 4
9 4
9 5
10 6
10 7
11 8
11 9
@find
3 B
@find
@end
```

Output for the Sample Input

```
none
8 11
```

Problem F

Battery of Electric Cars

Time Limit: 3 Seconds

GoodDog wants to collect data at n stations and there are only k transmission devices. The stations are connected by some roads, and GoodDog wants to build electric driverless cars to carry the transmission devices such that each car can collect the data of several stations by traveling between stations.

An important issue is to determine the capacity of the battery, and all the cars should have the same capacity. The cars can be recharged at any of the stations. To reduce the weight of the car, GoodDog hopes that the capacity of the battery is as small as possible. Of course, the capacity should be sufficient for the mission: collecting the data of all stations by using at most k transmission devices. That is, for each station, either there is a transmission device at the station or the station must be visited by a car. For the stations visited by the same car, there must be paths consisting of only roads required capacity c , where c is the battery capacity of the cars.

Given the required capacity of every road, your job is to determine the minimum capacity of the battery.

Technical Specification

1. Each integer between 0 and $n - 1$ represents one station, where $2 \leq n \leq 100000$ for each test case.
2. Each road connects two stations. The required capacity of any road is an positive integer at most 100000.
3. The number of roads m is a positive integer at most 500000.
4. $1 \leq k < n$.

Input File Format

The first line of the input file contains an integer T which indicates the number of test cases. Each test case starts with three integers n , m , and k in one line, where n , m , and k are the number of stations, roads, and transmission devices, respectively. Followed this line, there are m lines, and each line contains three integers u_i , v_i , and c_i , where u_i and v_i are the two stations the road connects and c_i is the required capacity to traverse the road. Two consecutive numbers in the same line are separated by a space.

Output Format

For each test case, output the minimum capacity in one line. If it is impossible, then output -1.

Sample Input

```
2
7 6 2
```

0 1 10
1 2 20
2 3 30
3 4 40
4 0 20
5 6 20
6 4 2
0 1 1
0 2 5
1 2 6
4 5 5

Output for the Sample Input

30
-1

Problem G

Secret Clubs

Time Limit: 3 seconds

There are two secret clubs, named X -club and Y -club. The X -club is formed by m college students $\{1, 2, \dots, m\}$, and the Y -club is formed by n college students $\{1, 2, \dots, n\}$. Any two students in the same club are friends. If two students are not friends, then they are from different clubs. Specifically, any student in one club has at least one non-friend student in the other club. For convenience, we use a *crossing-club-friendship* (i, j) to represent that student i in X -club and student j in Y -club are friends.

One day, the school administrators plan to make a survey of public opinion from the two clubs. They need to find a set S with the maximum cardinality such that any two students in S are friends. For example, consider $X = \{1, 2, 3\}$ and $Y = \{1, 2, 3\}$ and four crossing-club-friendships $(1, 2)$, $(1, 3)$, $(2, 2)$, and $(2, 3)$. In this case, we can select students 1 and 2 from X and students 2 and 3 from Y to form S . Hence, $|S| = 4$. Given m and n and k crossing-club-friendships, your task is to write a computer program to compute $|S|$.

Technical Specification

- $1 \leq m \leq 1000$.
- $1 \leq n \leq 1000$.
- $0 \leq k \leq \max\{m(n-1), n(m-1)\}$.

Input File Format

The first line of the input file contains an integer, denoting the number of test cases to follow. For each test case, the the first line contains three positive integers m, n , and k . In the following k lines, each line contains two positive integers separated by one space, which represent one crossing-club-friendship.

Output Format

For each test case, output $|S|$ in one line.

Sample Input

```
2
3 3 4
1 2
1 3
2 2
2 3
2 2 2
1 1
```

2 2

Output for the Sample Input

4

2

Problem H

Water Tanks and Hoses

Time Limit: *5 Seconds*

There are several water tanks placed at different levels as in Fig. 1. You have several hoses of different sizes and their lengths only allow you to connect water tanks between adjacent levels. A hose of throughput $1l/sec$ can transport 1 litre of water from a tank to a tank each second. In the beginning, the water tanks in top level are always full. At any time, there is one hose that can be installed. Once a hose is installed between two tanks, it cannot be unplugged when water is flowing. Only until the tank is empty, the hose can be unplugged and it cannot be used any more, i.e., thrown away. One water tank can only allow one hose to connect at a time.

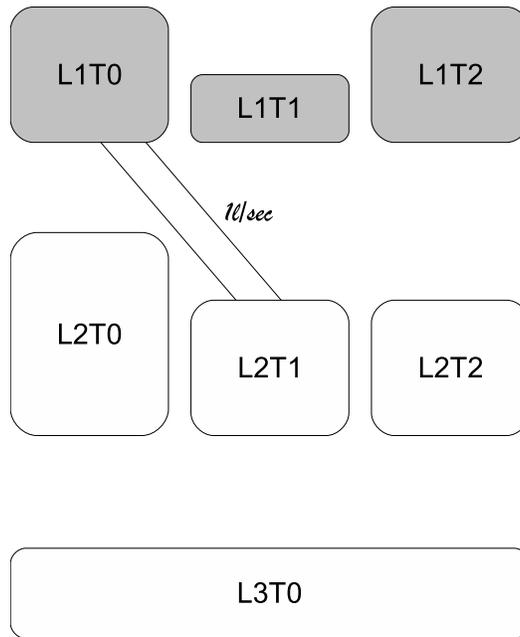


Figure 1: Level of tanks.

Assume the installation and unplugging of hoses cost 0 seconds. The water in the tanks are integer in litre. You cannot install a hose from tank x to tank y if the water in tank x overflows tank y . There is always only one large tank at the bottom level which is big enough to hold all the water.

Given levels of water tanks with top-level tanks filled with water and a set of hoses, please find a way to transport the water to bottom tanks with minimum time.

Input File Format

The test data begins with an integer N , which is the number of test cases. A test case begins with the levels of water tanks L ($L \leq 1000$). Following L is L lines of level data. For each level, it begins with the number of tanks T ($T \leq 6$) and then T integer numbers to represent the volume of water tanks. Following the level data is an integer H to represent the number of hoses. Then there are H floating numbers to indicate the throughput of hoses (litre/sec.).

Output Format

For each test case, please output the minimum time (in seconds) to transport all the water from top level to bottom level. If there is no solution, please output -1 for the test case. Please round off the decimals to two digits behind the point.

Sample Input

```
2
3
3 2 1 2
3 5 2 2
1 10
4
1 2 2 4
3
3 2 1 1
2 1 3
1 10
7
1 1 1 2 2 2 3
```

Output for the Sample Input

```
4.25
4.00
```