Southwest Jiaotong University

The 10$^{\text{th}}$ Collegiate Programming Contest

Final Round

12:15-17:15, Sunday, 20$^{\text{th}}$ April, 2014

Problem Set

Attention

1. Please don't use any other handle to review previous source code/submit your code, except the only specified handle to login and submit.
2. Please don't sumbit any malicious source code.
3. Please don't use any electronic devices, such as portable usb hard drives, calculators, mobile phones, etc.
4. Please don't cheat/exchange, together with other teams.
5. Please follow the contest rules.

Acknowledgement

I gratefully acknowledge the kind help of Mr. Hao for providing such suitable competition environment. I also gratefully acknowledge the financial support of our sponsor *Sichuan 517na inc.*

The problem set of this final round was recommended by *xiaoqian*, *huifang*, *tingting*, *lida*, *zhangkai* and *me*. Most were verified by *wangyue*, *taoran* and *me*. I approved all the problems. Thanks a lot for their powerful supports.

I acknowledge the kind help of *taoran*, *lingwei*, *xiangye*, *zhulei*, *yemeng*, *zhixin*, *guanjing*, *chengang etc.* for preparing the online contest. I acknowledge the kind help of students from ACM association of SWJTU during all the contest. I hope all of you enjoy this contest, thank you.

Problem A: Miss SOJ

Hee hee, er..., en..., glad to see you again, pretty girls and smart guys! My name is **SOJ**, which is short for *Southwest jiaotong university Online Judge*. Though now I am 10 years old, far younger than you, I claim to have been one of the oldest online judge system in China mainland. I am derived from the well-known **POJ**, which is short for *Peking university Online Judge*. Yes, you are right, POJ is my mother! Well, maybe you want to know whether or not I am a girl. Well, of course, you could have known it from the title. Okey, I still want to let you know that I am really a beautiful girl, like you or your girl friend, even I am now a **Cinderella**, hee hee!

To follow the law of **NATURE**, a big monster, I had to leave from my mother when I was a kid. Yes, indeed, I felt so sad when I came to *Southwest jiaotong university* as I really knew nothing about **it**. When I finally arrived at Xipu Capaus around the year 2004, there was no any decent buildings, flowers or plants. I imagined it only was **a dwarf poor**.

However, I found much happiness soon as a lot of teachers and students enjoyed programming with me. There were many professional teachers such as *Rong H. Zhou*, *Xiao F. Hao*, *Hai Y. Huang*, *F. Chen*, *Tian R. Li*, *Hong Y. Zhao*, *X. Gong*, *Shao J. Qiao* and *Xiu Y. Yu etc.*, that were really very concerned about me. Students such as *Yi L. Xiong*, *K. Zhang*, *Guo J. Fang*, *H. Huang*, *Jian P. Xu*, *Han Z. Wu*, *Ya Q. Mao* and *P. Du etc.*, enjoyed coding and algorithms with me once all the night.

I witnessed the moment we won our **First Bronze Medal** in the year 2008. We all were very exited about that moment and even burst into tears. Increasing students were then favor of programming and algorithms. We then won more and more medals in following years. For instance, we won the **First Prize** in **Topcoder** International Programming Challenge Open Contest (Sichuan Site), the **Gold Medal** in ACM International Collegiate Programming Contest Asia Regional Contest (Haerbin Site), the **First Place** in the 1^{st} Top Embedded Design C-Programming Contest, and the **Silver Medal** in "Google Cup" Nation Invitational Programming Contest (Fudan Univ.), etc. I liked to stay with them so that I changed my name "**POJ-XXX**" to "**SOJ**". Hee hee, likewise, nowdays, Xipu Capaus seems to be extremely beautiful.

Time Flies! Once enjoyed programming and algorithms together with me, most students now have already graduated from our *SWJTU*. Sometime, I feel some uncomfortable. I miss them, well, indeed, I find I love them deeply.

Well, it is certainly true that everyone has his or her great dream. I needn't to miss them now. But I am sure that those students once attracted by me (hee hee, er..., I don't like to say programming, how beautiful I am!) would benefit a lot. I wish they would maintain great achievements in the future! For me, I hope you would go on enjoying algorithms with me, wouldn't you? OK, from now, overpass our history described above, let's start with a programming task. *PS. NO? don't you like to make friends with me, a beautiful and clever girl?* OK, let's begin. Given two integers n and s, please find the biggest integer k such that s^k divides $n!$ without a remainder. $(n! = 1 \times 2 \times ... \times (n-1) \times n)$

Input

Input contains multiple cases. The first line contains an integer T $(1 \le T \le 100)$, indicating the number of cases. Each of the following T lines contains two integers n and s $(2 \le n \le 10^{18},\ 2 \le s \le 10^{12})$, representing one case.

Output

For each case, print the maximum integer k in a line.

Sample input

```
4
5 2
5 3
5 4
5 5
```

Sample output

```
3
1
1
1
```

Recommender

xiaoqian - 2006

Problem B: Longest Common Subsequence

Most of you are familiar with the topic of *longest common subsequence* (*LCS*), especially for those who have been *an acmer* for several years. Well, the LCS problem is to find the longest subsequence common to all sequence in a set of sequences (often just two). Note that a *subsequence* is different from a *substring*, for the terms of the former need not be consecutive terms of the original sequence. For example, for a sequence (2, 0, 1, 4), both (2, 0) and (0, 4) are one of its subsequences, while (1, 0) and (4, 0, 2) are not as *a subsequence* considers *order*. Now, your task is to determine the length of the longest common subsequence between two integer sequences.

Input

Input contains multiple cases. Each case starts with an integer n ($1 \leq n \leq 10^5$), indicating the length of both the two integer sequences. Each of the following two lines contains n integers, representing an integer sequence. It guarantees that all input integers will be in range [-10^9, 10^9], and no two identical integers are in the same sequence.

Output

For each case, print the length of the longest common subsequence in a line.

Sample input

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

Sample output

```
2
2
```

Recommender

huifang - 2004

Problem C: Find rectangles

Given an $N \times M$ grid, which has $N+1$ horizontal lines and $M+1$ vertical lines respectively. Please find the number of distinct rectangles, in which a rectangle consists of four distinct grid vertices.

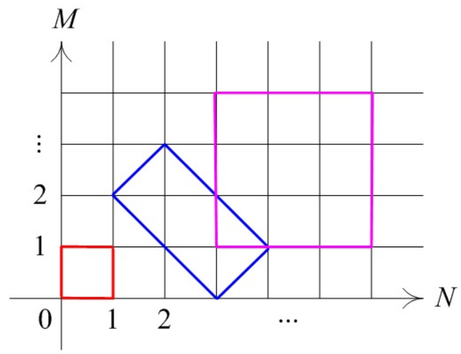

Figure 1: An example of finding rectangles.

Input

Input contains multiple cases (no more than 100), each of which contains two integers N and M ($1 \leq N, M \leq 10^3$), denoting the size of the grid.

Output

For each case, print the answer MOD 2014 in a line. Here, A MOD B means the remainder for B divides A, e.g. 3 MOD $1 = 0$, or 13 MOD $6 = 1$, and etc.

Sample input

```
1 1
3 3
```

Sample output

```
1
44
```

Problem D: Farmer John

If you once took part in USACO, which is short for USA Computing Olympiad, you will know who is *Farmer John*. Well, I am also sure that some of you know nothing about him. However, it does not matter. You could just help him solve programming tasks.

John has built a huge theme park for crops. Wow, he really looks like "*A Tall-Rich-Handsome*", and the park looks like "*A Gao-Da-Shang*". However, he has to face a thorny problem recently. He needs to build many irrigation equipments for the crops in the theme park. He should guarantee that each crop-site (denoted by a number) has an access to water source. More clearly, he could build a water equipment in a crop-site so that the crop-site can obtain water directly. And, any other crop-site that connects a crop-site containing a water equipment can also obtain the water indirectly via a water pipeline.

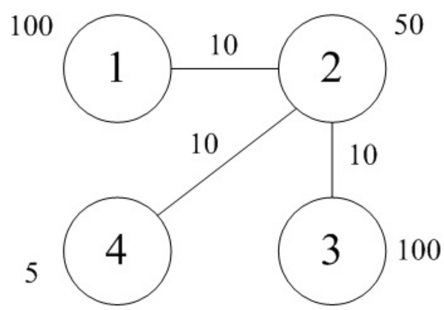

Figure 2: An example of crop-site map.

For example, Fig. 2 shows an example of crop-site map. As shown in Fig. 2, it contains 4 crop-sites, denoted by decimal number 1, 2, 3, and 4. An edge between two crop-sites indicates that a water pipeline can be built between them, otherwise, it is not able to build a water pipeline. For any two crop-sites that can be connected by a pipeline, it needs a certain cost to build the pipeline. For instance, if we want to build a pipeline between 1 and 2 (crop-site), we need a cost of 10, which is marked in Fig. 2. For any crop-site, if we want to build a water equipment (water source), we also need a certain cost. For instance, if we want to build a water equipment in 1 (crop-site), we need a cost of 100; if we want to build a water equipment in 4, we need a cost of 5, etc.

Farmer John hopes you help him to find the minimum cost to build all possible water equipments and pipelines so that each of the crop-sites can directly or indirectly obtain the water. E.g. for Fig. 2, we can build all the pipelines with a cost of 30 and build a water equipment in 2 with a cost of 50, this way, we need a total cost of 80. However, it is not a minimum cost! We can still build all the pipelines with a cost of 30, but build a water equipment in 4 with a cost of 5, this way, we only need a total cost of 35 and all the crop-sites can obtain water directly or indirectly.

Input

Input contains multiple cases. Each case starts with two integer n and m ($0 \le n, m \le 10^5$, $m \le \frac{n \times (n-1)}{2}$), indicating the number of crop-sites (numbered from 1, 2, ..., n) and the number of pipelines that could be built, respectively. The following line contains n non-negative integers, where the k^{th} integer $c_k (0 \le c_k \le 10^9)$ represents the cost of building a water equipment in the k^{th} crop-site. Then each of the following m lines contains three integers $u_k, v_k, w_k (1 \le k \le m, u_k \ne v_k, 1 \le u_k, v_k \le n, 0 \le w_k \le 10^9)$, indicating a cost of w_k to build a pipeline between u_k and v_k. It guarantees that at most one pipeline could be built for any two crop-sites.

Output

For each case, print the minimum cost in a line.

Sample input

```
4 3
100 50 100 5
1 2 10
2 3 10
2 4 10
```

Sample output

```
35
```

Hint

The sample case corresponds to the example given in the description.

Problem E: Ticket and balance

You are to deal with two sub-problems sponsored by *Sichuan 517na inc.*

1. TICKET

In order to encourage traveling by plane, some airline decides to sell some tickets with a relative cheaper price in weekends (i.e. Saturday and Sunday). Namely, a ticket is 800 *yuan* in weekends and 1000 *yuan* in other days. Suppose you ready to travel after a^b days from today (Sunday) according to this airline. What will be your own ticket price?

2. BALANCE

Back from a happy trip, you bring n lovely gifts back to your friends. When you arrive at home from the airport, you find a balance on your desk. "Haoba", only with the gifts and the balance, you want to compute the number of different positive weights that you could weigh. You also want to know the maximum one that you could weigh. "Haoba", how boring!

Figure 3: An example of a balance.

Input

Input contains multiple cases (≤ 10). Each starts with two integers a, b ($1 \leq a,\ b \leq 10$). Then the following comes an integer n ($1 \leq n \leq 10$), indicating the number of gifts. Then the following line contains n positive integers w_k ($1 \leq k \leq n, 1 \leq w_k \leq 10^9$), denoting the relative weights of the n gifts.

Output

Please print the price of your ticket, the number of different positive weights as well as the maximum one in a line (separated by a blank) for each case.

Sample input

```
1 1
2
1 2
1 1
1
1
```

Sample output

```
1000 3 3
1000 1 1
```

Hint

For the 1^{st} case, there are 2 gifts with a weight of 1 and 2 respectively. It can be easily determined that 3 different positive weights can be weighed out, namely 1, 2 and 3. And the maximum one is 3.

Recommender

Sichuan 517na inc.

Problem F: Sudoku

I think most of you konw how to play Sudoku on an $N \times N$ board, in which cells are grouped in $M \times M$ squares, where M^2 always equals N. Each cell is either empty or contains an integer ranged $[1, N]$. The puzzle game is finished when integers in each row, each column and each square are different. The most familiar Sudoku is played on a board of 9×9 cells, as the following figure.

Figure 4: A puzzle game of Sudoku played on a board of 9×9 cells.

Actually, we could also play Sudoku on a strange board as shown in following figure. It seems to be more difficult and interesting. However, now your task is not to deal with such a strange board as it is prepared for some of you in this summer. And now, your problem is relative simple, namely to judge whether the goal of Sudoku played on an $N \times N$ board can be addressed. Here, the goal is to fill all empty cells so that all integers in each row, each column and each square are different finally when given an initial situation.

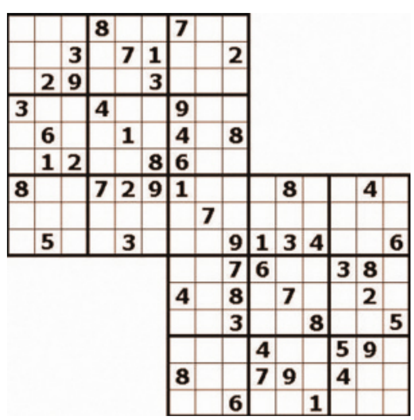

Figure 5: An example of Sudoku played on a strange board.

Input

Input contains multiple cases. Each case starts with an integer N where it meets that $N = M^2, 1 \le M \le 4$. Then each of the following N lines contains N integers, indicating the initial situation, in which the j^{th} integer in the i^{th} line denotes the value $a_{i,j}$ $(1 \le i, j \le N)$ in the corresponding position. Note that the corresponding cell is empty if $a_{i,j} = 0$, otherwise it indicates the value of the corresponding position. We guarantee that all integers are ranged $[1,\ N]$.

Output

For each case, print "Yes" if the goal can be addressed, otherwise print "No".

Sample input

```
9
5 2 0 0 0 6 0 0 0
0 0 0 0 0 0 7 0 1
3 0 0 0 0 0 0 0 0
0 0 0 4 0 0 8 0 0
6 0 0 0 0 0 0 5 0
0 0 0 0 0 0 0 0 0
0 4 1 8 0 0 0 0 0
0 0 0 0 3 0 0 2 0
0 0 8 7 0 0 0 0 0
```

Sample output

```
Yes
```

Hint

For the sample case, we could construct the goal as follows:

```
5 2 7 3 1 6 4 8 9
8 9 6 5 4 2 7 3 1
3 1 4 9 8 7 5 6 2
1 7 2 4 5 3 8 9 6
6 8 9 2 7 1 3 5 4
4 5 3 6 9 8 2 1 7
9 4 1 8 2 5 6 7 3
7 6 5 1 3 4 9 2 8
2 3 8 7 6 9 1 4 5
```

Problem G: Who is the cracker

As we all know, there are lots of hackers lurking in the network. Some of them, to whom the name cracker is more fit, may hack into a work station, a web server, or a personal computer etc., and make some damages. So, in accordance with the law, hacking into a computer without permission is criminal. Snowden is a cyber police, whose task is to monitor the network and finding out who is a suspect of cyber criminal. It seems that he got into trouble with this job recent days. Can you help him to find out the suspect of crackers?

To simplify the problem, we abstract the network as an undirected graph in which vertexes respresnt computers and edges resprent cables. If using computer A to attack computer B, a cracker will choose some computers to establish a chain from A to B, to avoid inspection or network tracing. Now, we define a value p of the edge between i and j, which means it is at risk of establishing a link between them, and the probability of being detected is p‱. Unfortunately, the crackers are so clever that they will always choose a proper chain to minimize the risk. In other word, whether a cracker be discovered or not depends on the maximum value among all the risk-values of edges in the selected chain (i.e. the selected path).

In addition, considering that we are dealing with a real-time environment, the cable between computer i and j may break down due to various reasons. It will be unable to use until repaired.

Input

Input contains a single case. In this case, the first line contains three integers n, m, q, where n $(1 \le n \le 10^5)$ is the number of computers, m $(1 \le m \le 10^6)$ is the number of edges between computers, and q $(1 \le q \le 10^5)$ indicates there are q operations in this case. Each of the following m lines contains three integers i, j and v $(1 \le i, j \le n,\ 0 \le v \le 10^4)$, representing an edge between i, j with a risk-value of v. Then q lines comes in the following, each of which contains three integers k, i and j $(i \ne j, 1 \le i, j \le n)$, indicating

- $k = 1$, a cracker try to establish a chain (i.e. some path) from i to j.
- $k = 2$, the cables directly connected between i and j will be broken down.

Input data guarantees that the path from i to j exists in case $k = 1$, and the cable between i and j exists in case $k = 2$.

Output

For each $k = 1$, print the minimum risk among all possible chains in a line.

Sample input

```
4 4 3
```

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

Sample output

```
2
3
```

Hint

Input is huge and will take too much time for reading in. Therefore, for C/C++ users, we recommend using the function below to read an integer:

```
typedef int type_t;
type_t readin_an_integer()
{
  for(char ch = getchar(); ch > ‘9’ || ch < ‘0’; ch = getchar());
  for(type_t tmp = 0; ‘0’ <= ch && ch <= ‘9’; ch = getchar())
    tmp = tmp * 10 + (ch - ‘0’);
  return tmp;
}
```

Recommender

lida - 2010

Problem H: MMs

M MMs are standing on the vertices of a tree with M vertices numbered from 1 to M. Each vertex is either empty or contains a lot of MMs. It would make MMs unpleased when one vertex is shared by many MMs. So the task is to let MMs move along the edges between vertices so that each vertex has exactly one MM finally. We denote each move for MMs by moving one MM in her located vertex to its adjacent vertex. However, it needs a certain cost to move from one vertex to its adjacent vertex. Therefore, please tell me the minimum cost to achieve the goal that each vertex has exactly one MM according to a number of moves.

Input

Input contains multiple cases. Each case starts with an integer M ($1 \leq M \leq 10^5$), denoting the number of MMs as well as the number of vertices in the tree. The following M-1 lines represent the edges (including its cost) of the tree, each of which contains three integers u_i, v_i, c_i ($1 \leq i \leq M-1,\ 1 \leq u_i \neq v_i \leq M, 1 \leq c_i \leq 10^9$), indicating there is an edge between vertex u_i and v_i with a cost of c_i. Then the following line contains M integers a_i ($1 \leq i \leq M$) in order, indicating the number of MMs in the i^{th} ($1 \leq i \leq M$) vertex. It guarantees that $\sum_{i=1}^{M} a_i = M$.

Output

For each case, print the minimum cost in a line.

Sample input

```
1
1
2
1 2 3
2 0
```

Sample output

```
0
3
```

Recommender

xiaoqian - 2006

Problem I: Points on squares

Considering n different points on the 2-D plane, you need to draw a lot of squares with edges parallel to x- and y- axis so that each point should be located at the middle of the top or bottom edges in a square. All the squares should have the same size, and no two squares overlap, except that they could share one edge. The following figure gives an example of finding such suitable squares. Now, your task is to find the maximum integer square size such that all points are located at the top or bottom edges of the squares.

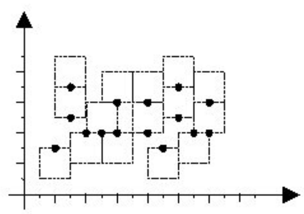

Figure 6: An example of finding such suitable squares.

Input

Input contain multiple cases (no more than 12). Each case starts with a line containing an integer n ($3 \le n \le 100$), indicating the number of points. Each of the following n lines contains two integers $x_i,\ y_i$ ($1 \le i \le n, -10^4 \le x_i,\ y_i \le 10^4$), denoting the coordinate of the i^{th} point. Note that no two points have the same coordinate.

Output

For each case, print the maximum possible square size (an integer) in a line.

Sample input

```
3
1 0
2 0
3 0
```

Sample output

```
2
```

Recommender

tingting - 2005

Problem J: Hapsunday

Hi, everyone, hapsunday! Invited by *hanzhou*, once we worked together for *icpc*, I am really pleased to recommend a simple problem to celebrate such a great programming contest for all *swjtuers* and *acmers* from fraternal universities! Okay, let's come up with *a hapsunday problem*, I guess you would quickly slove it, don't you? please find the value of $hapsunday(n) = \sum_{k=1}^{n} \frac{k}{(k+1)!}$.

Input

Input contains multiple cases. Each case contains an integer n $(1 \leq n \leq 64)$.

Output

For each case, print the real value with 2 digits after the decimal point in a line.

Sample input

```
1
```

Sample output

```
0.50
```

Hint

$n! = 1 \times 2 \times 3 \times ... \times (n-1) \times n,\ (n \geq 1)$.

Recommender

zhangkai - 2006

Problem K: Minimum steps

Starting with 1 by only adding 1, we could compute 31 with 30 steps:

$$2 = 1 + 1,\ 3 = 2 + 1,\ \ldots,\ 31 = 30 + 1.$$

The "double" operation can be used to reduce the number of steps (8 steps):

$$2 = 1 + 1,\ 3 = 2 + 1,\ 6 = 3 + 3,\ 7 = 6 + 1,$$
$$14 = 7 + 7,\ 15 = 14 + 1,\ 30 = 15 + 15,\ 31 = 30 + 1.$$

This is not the minimum one as:

$$2 = 1+1,\ 4 = 2+2,\ 8 = 4+4,\ 10 = 8+2,\ 20 = 10+10,\ 30 = 20+10,\ 31 = 30+1.$$

If subtraction is also available, it can be further reduced with only 6 steps:

$$2 = 1 + 1,\ 4 = 2 + 2,\ 8 = 4 + 4,\ 16 = 8 + 8,\ 32 = 16 + 16,\ 31 = 32 - 1.$$

Your task is to compute the minimum steps to get n by only using addition and subtraction starting with 1 for the given positive integer n. During each step, such values $-k$ ($k \leq 0$) should never appear. You can use k ($k \geq 1$) that you have obtained from previous steps during computing process.

Input

Input contains multiple cases. Each only contains an integer n ($1 \leq n \leq 10^3$).

Output

For each case, print the minimum steps to get n in a line.

Sample input

```
31
473
953
```

Sample output

```
6
11
12
```

Hint

The corresponding sequences for sample cases are:
(2, 4, 8, 16, 32, 31),
(2, 4, 8, 16, 32, 31, 63, 126, 252, 504, 473),
(2, 4, 8, 7, 15, 30, 60, 120, 240, 480, 473, 953).