Problem A
Right-Coupled Numbers

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

An integer x is said to be a right-coupled number, if you can find two integers, say $0 < a \leq b \leq x$ such that $a \times b = x$ and $a/b \geq 0.5$. In this problem, your task is to determine whether a given integer is a right-coupled number or not.

Input Format

The first line of the input is an integer N denoting the number of test cases. Each test case is in one line, which contains a single integer $0 < x < 2^{15}$.

Output Format

If the given integer x is a right-coupled number, output 1; otherwise, output 0. Each is in a single line.

Technical Specification

- $1 \leq N \leq 1000$
- $0 < x < 2^{15}$

Sample Input 1

```
4
66
55
105
150
```

Sample Output 1

```
1
0
0
1
```

Almost blank page

Problem B
Make Numbers

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

Peter is a math teacher at an elementary school. To familiarize students with three basic arithmetic operations plus (+), minus (−) and times (×), he gives a simple arithmetic puzzle as homework. The puzzle is that you are given 4 digits, and you are told to build as many non-negative integers as possible using just those 4 digits and at least one of the three basic arithmetic operations. For example, you are given 1,1,2,1 as the digits, and then you can build 32 non-negative integers as Table 1.

Table 1: Numbers made by 1,1,2,1.

$0 = 2 - 1 - 1 \times 1$	$22 = 21 + 1 \times 1$
$1 = 2 + 1 - 1 - 1$	$23 = 21 + 1 + 1$
$2 = 2 + 1 - 1 \times 1$	$32 = 21 + 11$
$3 = 2 + 1 + 1 - 1$	$109 = 111 - 2$
$4 = 2 + 1 + 1 \times 1$	$111 = 112 - 1$
$5 = 2 + 1 + 1 + 1$	$112 = 112 \times 1$
$8 = 11 - 2 - 1$	$113 = 112 + 1$
$9 = 11 - 2 \times 1$	$120 = 121 - 1$
$10 = 12 - 1 - 1$	$121 = 121 \times 1$
$11 = 12 - 1 \times 1$	$122 = 121 + 1$
$12 = 12 + 1 - 1$	$132 = 12 \times 11$
$13 = 12 + 1 \times 1$	$210 = 211 - 1$
$14 = 12 + 1 + 1$	$211 = 211 \times 1$
$19 = 21 - 1 - 1$	$212 = 211 + 1$
$20 = 21 - 1 \times 1$	$222 = 111 \times 2$
$21 = 21 + 1 - 1$	$231 = 21 \times 11$

To check whether the student's answer includes all possible integers, Peter needs to know the total number of non-negative integers that can be built for the puzzle. Please write a program to help Peter compute the total number.

Input Format

The input file contains 4 integers separated by a space in a line, which indicates the given digits.

Output Format

Output the total number of non-negative integers that can be built.

Technical Specification

- The expressions are composed by concatenating the 4 given digits and at least one operation in $\{+, -, \times\}$. The given digits are the elements in $\{1, 2, 3, \ldots 9\}$.
- The given digits are partitioned into several groups and the digits in each group are concatenated as a number in arbitrarily permutation order.
- The symbol $-$ can only be treated as a minus operator.
- The operations $+$ and $-$ have equal precedence.
- The operation $\times$ has higher precedence than $+$ and $-$.
- Operations with the highest precedence are evaluated first, and operations with equal precedence are evaluated from left to right.

Sample Input 1

```
1 1 1 1
```

Sample Output 1

```
15
```

Sample Input 2

```
1 1 2 1
```

Sample Output 2

```
32
```

Problem C
Pyramid

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Consider an $n \times n$ grid where nodes are labelled as (i, j) for $0 \le i, j < n$. We rotate it 45 degree in clockwise direction and keep only its top half part. Then you get a *pyramid*, as shown in Figure 1. All of the nodes laid on the anti-diagonal of the grid have labels $(n - 1 - j, j)$ for $0 \le j < n$. They are located at the bottom line of the pyramid. For simplicity and clarity, we name node $(n - 1 - j, j)$ as exit j. Node $(0, 0)$ is called the starting point (labelled as P in Figure 1). When you insert a ball from the starting point, this ball will roll down to some of the exits. A ball located at node (i, j) can only roll down to either node $(i + 1, j)$ or node $(i, j + 1)$, and the ball shall never be broken or split. There is a tiny switch equipped on every node other than the exits that controls the move direction of the ball, and this switch always works well. The switch has exactly two states: either L or R, indicates that the ball can move to node $(i + 1, j)$ or $(i, j + 1)$, respectively. After the ball leaves this node, the switch changes immediately to the other state. The default setting for a switch is at L.

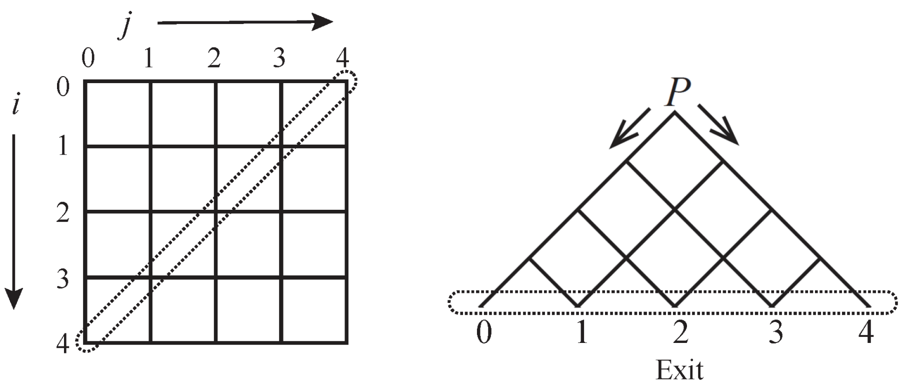


Figure 1: An example for $n = 5$.

When you insert the first ball into P, this ball will reach exit 0, and the states of nodes $(i, 0)$ for $0 \le i < n - 1$ are now all R's. Then you insert the second, third, and so on so forth, one by one, until the k^{th} ball finishes. The status of every switch accumulates with these balls. Please write a program to determine the number of the exit point for the k^{th} ball.

Input Format

The first line of the input is a number that specifies the number of test cases. Each test case has only one line that gives you two space-delimited numbers n and k.

Output Format

Please output the exit number of the k^{th} ball in one line.

Technical Specification

- There are at most 20 test cases.
- $1 \leq n \leq 10^4$.
- $1 \leq k \leq 10^8$.

Sample Input 1

```
2
5 1
5 2
```

Sample Output 1

```
0
1
```

Sample Input 2

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

Sample Output 2

```
2
3
2
```

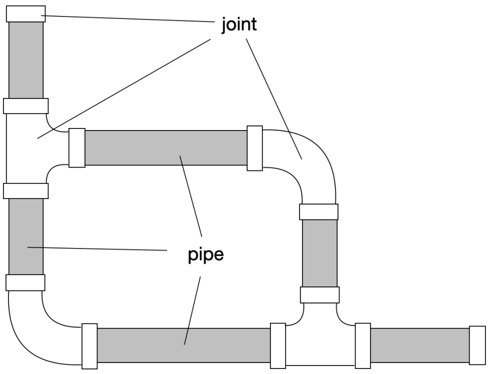
Problem D
Quality Monitoring

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

To provide a better drinking quality, the government is going to deploy some "smart devices" into the water supplying system so that the quality of the water can be monitored. The water supplying system consists of many pipes, and two pipes are connected by a joint. You may assume that the system forms a **connected simple graph**, with pipes being the edges and joints being the vertices. An example is given in the following figure.



The smart devices are designed to be deployed at the joints. However, two adjacent devices may interfere with each other, so it is required that no two devices are adjacent. There have to be enough number of devices deployed so that the system can be **fully monitored**. Formally, the system is fully monitored if

- there are at least $n - 28$ devices deployed, and
- no two devices are adjacent.

Please determine whether the system can be fully monitored. If the answer is no, output -1; otherwise, output the maximum number of devices that can be deployed.

Input Format

The first line of the input file contains two positive integers n and m, where n is the number of joints, numbered from 0 to $n - 1$, and m is the number of pipes. Each of the following m lines contains two nonnegative integers, indicating the joints at two ends of a pipe.

Output Format

Output an integer: "`-1`" if the system cannot be fully monitored; otherwise, the maximum number of devices that can be deployed.

Technical Specification

- $2 \le n \le 3 \times 10^4$, $1 \le m \le 5 \times 10^5$

Sample Input 1

```
5 7
1 0
2 3
1 4
1 2
3 1
3 4
0 4
```

Sample Output 1

```
2
```

Problem E
A Color Game

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Playing games is fun. For programmers, however, playing games with programs is even more fun. Consider a simple single-user tabletop game as follows. Given a row of sticks, each of which is in one of the seven colors, red (`R`), green (`G`), blue (`B`), cyan (`C`), magenta (`M`), yellow (`Y`), and key (`K`), the goal of the game is to eliminate all the sticks by repeating the following rules.

- Consecutive sticks with the same color can be eliminated if the size of them is not less than m.
- The remaining sticks will move closer together.

For the case where the row is `BBBRRRRRRGGGB` and m is 3, all the sticks can be successfully eliminated as the following steps:

1. `BBBRRRRRRGGGB`
2. `BBBGGGB` (By eliminating all red sticks)
3. `BBBB` (By eliminating all green sticks)
4. (By eliminating all blue sticks)

For the same row of sticks with $m = 4$, however, it is no way to eliminate all the sticks.

Given a row of n sticks and the value of m, your task is to determine if it is possible to eliminate all the sticks.

Input Format

Each test case is given as a string that is the row of sticks and an integer m.

Output Format

Output `Yes` if it is possible to eliminate all the sticks. Otherwise, output `No`.

Technical Specification

- $0 < n, m \leq 500$

Sample Input 1

```
BBBRRRRRRGGGB 3
```

Sample Output 1

```
Yes
```

Sample Input 2

```
BBBRRRRRRGGGB 4
```

Sample Output 2

```
No
```

Problem F
Cable Protection

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

A company ICPC (International Cable Protection Company) produces a cable protection tool that can be installed in a network switch to monitor whether all cable links connected to it are working properly. Because the protection tool would cause transmission delay, it is not suitable for installation on every switch.

Usually network topology consists of two parts: a backbone and several subnets. The switches on the backbone are linked as a ring structure and each backbone switch is treated as a root of a subnet in which the switches are linked as a tree structure. We call such topology as unicyclic topology. Figure 2 shows an example of a unicyclic topology.

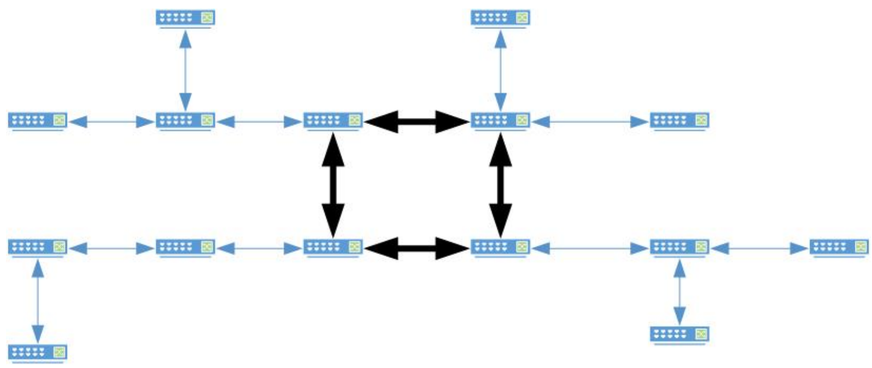


Figure 2: An example of uncyclic topology.

Suppose there are n backbone switches and m subnet switches. The switches are numbered by integers from 0 to $m + n - 1$. Backbone switches are numbered from 0 to $n - 1$ in clockwise order and the subnet switches are numbered from n to $n + m - 1$ where the index of each subnet switch is larger than the index of its parent in the rooted tree structure of the subnet it belongs. Figure 3 shows an example of switch numbering.

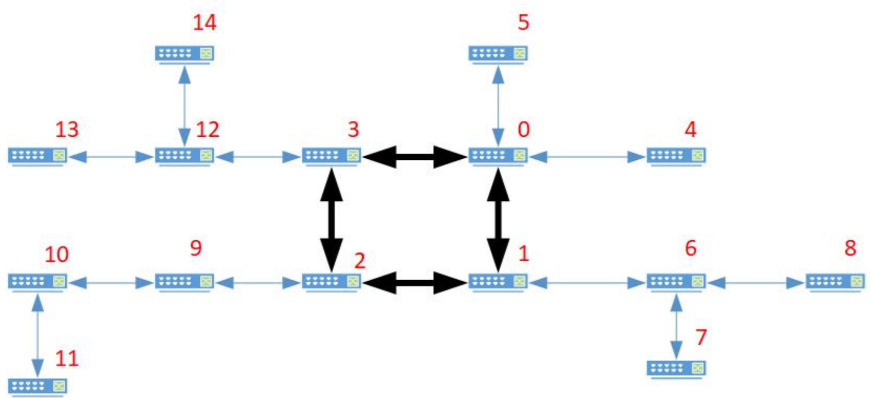


Figure 3: An example of switch numbering.

Please write a program for ICPC to decide the minimum number of switches selected for installing cable protection tools that can monitor all the cable links. Figure 4 shows an optimum solution (circled by ellipses) for the given network.

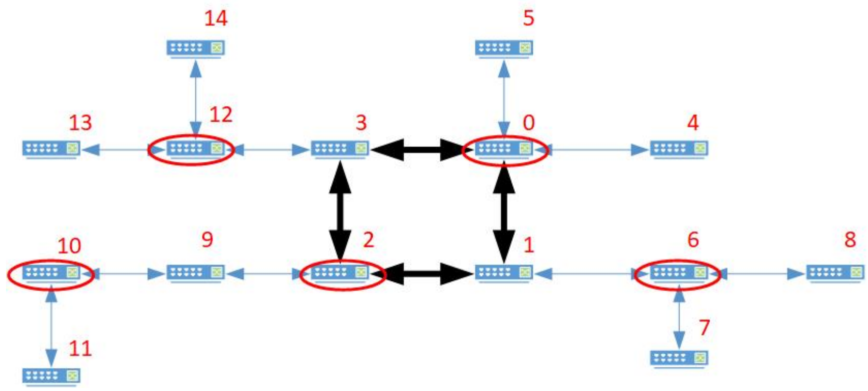


Figure 4: An optimum solution for the given network.

Input Format

The first line of the input file contains two integers n and m, separated by a space, indicating the numbers of backbone switches and subnet switches respectively. Each of the next $n+m$ lines consists of two integers, separated by a space, indicating the indices of the two end switches of a link.

Output Format

Output the minimum number of switches selected for installing cable protection tools that can monitor all the cable links.

Technical Specification

- $3 \leq n \leq 100000$
- $1 \leq m \leq 100000$

Sample Input 1

```
3 2
0 1
1 2
0 2
1 3
2 4
```

Sample Output 1

```
2
```

Sample Input 2

```
4 11
0 1
0 3
0 4
0 5
1 2
1 6
2 3
2 9
3 12
6 7
6 8
9 10
10 11
12 13
12 14
```

Sample Output 2

```
5
```

Almost blank page

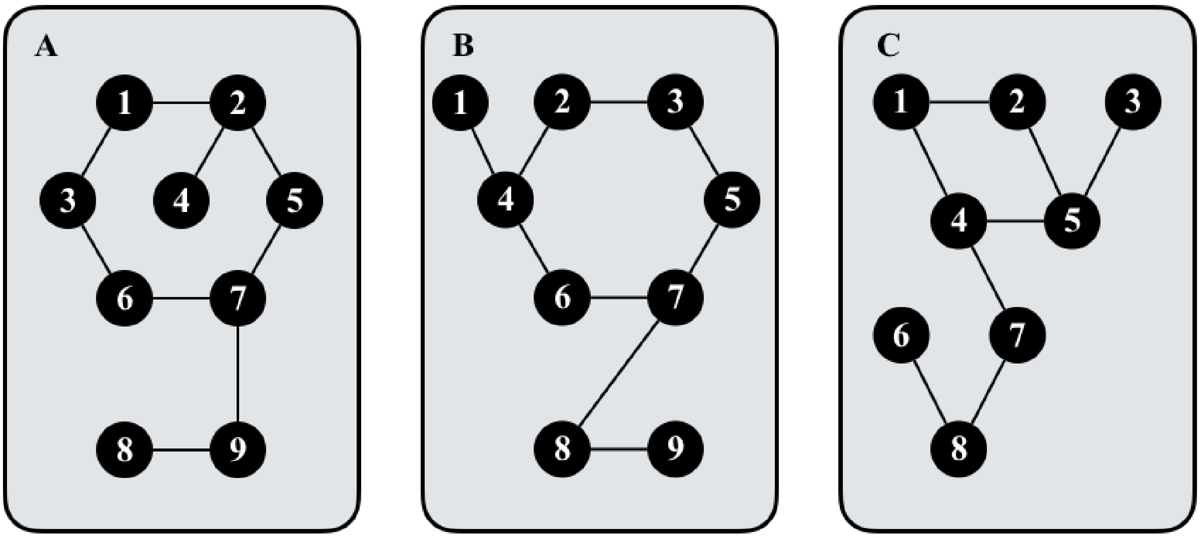
Problem G
Graph Cards

Time limit: 4 seconds

Memory limit: 1024 megabytes

Problem Description

A deck of graph cards is placed on the table. Each graph card χ is decorated with an undirected simple graph G_χ so that G_χ is connected and G_χ has the same number of nodes and edges. Note that different graph cards may have different numbers of nodes. An example is depicted as follows.



We say two graph cards are identical if and only if the graphs associated with them, say $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, are ***isomorphism***; that is, there exists a bijection f between the node sets V_1 and V_2 so that for every $x, y \in V_1$, edge $(x, y) \in E_1$ if and only if edge $(f(x), f(y)) \in E_2$. Our goal is to compute the number of distinct graph cards in the deck.

Input Format

The first line contains an integer t that indicates the number of test cases. For each test case, you are given a deck of graph cards. It begins with a line containing the number of graph cards $n > 0$. Then, n lines follow. Each line represents a graph card associated with a graph G in the following format:

$$k \quad u_1 \quad v_1 \quad u_2 \quad v_2 \quad \cdots \quad u_k \quad v_k$$

where $k > 0$ denotes the number of nodes (also edges) in G and for each $i \in [1, k]$ (u_i, v_i) denotes an edge in G that connects nodes u_i and v_i. Note that the identifiers of nodes are integers in $[1, k]$.

Output Format

For each test case, output the number of distinct graph cards in the given deck on a line.

Technical Specification

- $0 < t \leq 3$
- $0 < n, k \leq 10^6$.
- For each test case, the numbers of nodes in the n graph cards sum up to at most 10^6.

Sample Input 1

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

Sample Output 1

```
1
```

Sample Input 2

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

Sample Output 2

```
2
2
```

Problem H
Optimization for UltraNet

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

The UltraNet company provides network connectivity for all cities in a country. For a pair of cities, they are either directly connected or indirectly connected. A city i and another city j are directly connected if a cable with a symmetrical bandwidth of $b_{i,j} = b_{j,i}$ is wired between them. For two cities that are not directly connected, at least one path between the two cities exists. In the current UltraNet, more than one path is possibly available for a city pair.

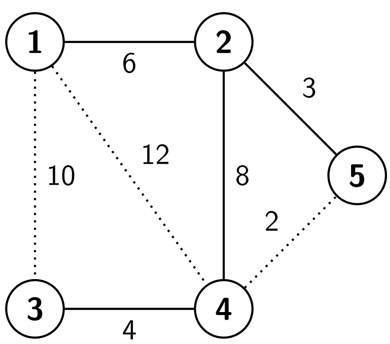
The current UltraNet is perfectly working, while the maintenance fee of each cable is costly. Energy conservation is another concern. The energy consumption of a cable is proportional to its bandwidth. Therefore, the company has an optimization plan to reorganize the network with the policies in the following order:

1. The number of cables should be minimized without sacrificing the connectivity of the whole UltraNet. In other words, exactly one path between every city pair should be satisfied.

2. If there is more than one way to minimize the number of cables, the bottleneck of the whole network should be maximized. The bottleneck of a network is determined by the city pair with the narrowest bandwidth, and the bandwidth of a city pair (i, j), $b'_{i,j}$, is determined by the cable with the narrowest bandwidth on the only path from i to j.

3. If there is still more than one way to meet the above two points, the total energy consumption of the network should be minimized. In other words, the sum of bandwidths of the remaining cables should be minimized.

Your task is to help UltraNet optimize the network and then output the sum of the bandwidths among all city pairs in the optimized network. For optimizing the following example, the three cables in dotted will be discarded. In the resulting network, the bottleneck is 3, the sum of bandwidths of the remaining four cables is $6 + 3 + 8 + 4 = 21$, and the sum of the bandwidths among all city pairs is $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b'_{i,j} = 6 + 4 + 6 + 3 + 4 + 8 + 3 + 4 + 3 + 3 = 44$.



Input Format

Each test case begins with two integers n and m, denoting numbers of cities and cables, respectively. Then, m lines will follow for specifying the m cables. Each of the m lines contains three positive integers, i, j, and $b_{i,j}$, denoting that a cable with a bandwidth of $b_{i,j}$ directly connects the city pair (i, j), where the cities are numbered from 1 to n, and $i < j$ since $b_{i,j} = b_{j,i}$. No more than one cable will be available between every city pair in the current network. In addition, the bandwidths of all cables are distinct; no two cables have the same bandwidth rating.

Output Format

The output is a single integer that is the sum of the bandwidths of all city pairs in the optimized network.

Technical Specification

- $2 \le n \le 10000$
- $1 \le m \le 500000$
- $1 \le i < j \le n$
- $0 < b_{i,j} < 10^7$

Sample Input 1

```
3 3
1 2 5
1 3 6
2 3 8
```

Sample Output 1

```
20
```

Sample Input 2

```
5 7
1 2 6
1 3 10
1 4 12
2 4 8
2 5 3
3 4 4
4 5 2
```

Sample Output 2

```
44
```

Sample Input 3

```
5 5
2 5 1
1 2 2
2 3 4
1 3 5
2 4 6
```

Sample Output 3

```
24
```

Almost blank page

Problem I
Critical Structures

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Intelligence Cloud Privacy Company (ICPC) is a world famous cloud service company that aims at developing secure and powerful cloud computing environments for users. Engineers in the ICPC construct a data center with n computing nodes, denoted by $1, 2, \ldots, n$, and m communication links. We can model this data center as an undirected graph $G = (V, E)$, in which n vertices represent n computing nodes and an edge between Node i to Node j if there is a communication link between them; we also call i and j are two end-nodes of this link. In addition, for two arbitrary nodes i and j in G, there is at most one communication link between i and j, and there is no communication link between the same node.

A *linear array structure* in a data center G is a sequence of nodes $\langle v_0, v_1, \ldots, v_{k-1}\rangle$, where $k \geq 2$, such that any two consecutive v_{i-1} and v_i for $1 \leq i \leq k-1$ have a communication link, and v_i for $0 \leq i \leq k-1$ are all distinct. A *ring structure* is a sequence of nodes $\langle v_0, v_1, \ldots, v_{k-1}\rangle$, where $k \geq 4$, such that any two consecutive v_{i-1} and v_i for $1 \leq i \leq k-1$ have a communication link, $v_0 = v_{k-1}$ and v_i for $0 \leq i \leq k-2$ are all distinct. A data center G is *connected* if there is a linear array between any two nodes; otherwise, it is *disconnected*. Using some elegant resource allocation algorithm, a research team of the ICPC needs to find the following critical structures for enhancing the privacy and security:

1. Critical node: a node in G whose removal disconnects G.
2. Critical link: a communication link in G whose removal disconnects G.
3. Critical component: a maximal set of communication links in G such that any two communication links in the set lie on a common ring.
4. Largest critical component: a critical component with the maximum number of communication links.

Given a connected data center G, your task is to write a computer program for computing the number of critical nodes, the number of critical links, and

$$\mu^* = \frac{\text{the number of critical components}}{\text{the number of communication links in a largest critical component}} = \frac{p}{q},$$

where $\frac{p}{q}$ is an irreducible form of μ^*.

Input Format

The first line of the input file contains an integer L ($L \leq 10$) that indicates the number of test cases as follows. For each test case, the first line contains two integers (separated by a space) representing n and m. Then it is immediately followed by m lines, in which each line contains two integers that represent two end-nodes of a communication link; moreover, any two consecutive integers are separated by a space.

Output Format

The output contains one line for each test case. Each line contains four positive integers representing the number of critical nodes, the number of critical links, p, and q, where $\frac{p}{q}$ is an irreducible form of μ^*. Note that any two consecutive integers are separated by a space.

Technical Specification

- $3 \leq n \leq 1000$ for each test case.
- $n - 1 \leq m \leq \frac{n(n-1)}{2}$.
- The sum of m in all L tests is smaller than 10^6.

Sample Input 1

```
1
6 6
1 2
2 3
3 4
4 5
5 6
6 1
```

Sample Output 1

```
0 0 1 6
```

Sample Input 2

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

Sample Output 2

```
2 1 1 1
```

Almost blank page

Problem J
Puzzle Game

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

For a string S, define $Adjacency(S)$ to be the *multiset* of *unordered* pairs $(S[i], S[i+1]), i = 1, 2, ..., |S| - 1$, and define $\Sigma(S)$ to be the multiset of $S[i], i = 1, 2, ..., |S|$, where $|S|$ is the length of S and $S[i]$ is the ith character of S. For example, for $S =$ ABADDADCAB, we have $Adjacency(S) = \{$AB, BA, AD, DD, DA, AD, DC, CA, AB$\} = \{$AB, AB, AB, AC, AD, AD, AD, CD, DD$\}$ and $\Sigma(S) = \{$A, A, A, A, B, B, C, D, D, D$\}$.

John is playing a puzzle game, in which two strings P and Q, $|P| > |Q|$, over the character set $\{$A, B, C, D$\}$ are given and the goal is to insert characters into Q to obtain a string Q' such that $\Sigma(Q') = \Sigma(P)$ and $Adjacency(Q') = Adjacency(P)$. For example, given $P =$ ABADCAB and $Q =$ CBB, by inserting A, D, A, A into Q, we can obtain a string $Q' =$ $\underline{\text{AD}}$C$\underline{\text{A}}$B$\underline{\text{A}}$B, in which inserted characters are underlined. It is easy to check that $\Sigma(Q') = \Sigma(P) = \{A, A, A, B, B, C, D\}$ and $Adjacency(Q') = Adjacency(P) = \{AB, AB, AB, AC, AD, CD\}$. Thus, Q' is a solution for $P =$ ABADCAB and $Q =$ CBB. As another example, for $P =$ ABA and $Q =$ CB, there is no solution.

Please write a program to help John. More specifically, given two strings P and Q, your program computes a string Q' such that Q' is obtained from Q by inserting some characters, $\Sigma(Q') = \Sigma(P)$, and $Adjacency(Q') = Adjacency(P)$.

Input Format

The first line of the input is an integer t, indicating that there are t test cases. Each test case consists of three lines: the first gives two integers, indicating the lengths $|P|$ and $|Q|$, the second gives the string P, and the third gives the string Q.

Output Format

For each case, output a solution string Q'. If there are multiple solutions, you can output any of them. If there is no solution, output "NO".

Technical Specification

- The number of test cases is at most 15.
- The length of P, $|P|$, is an integer between 2 and 10^3.
- The length of Q, $|Q|$, is an integer between 1 and 10^3 and $|P| - 18 \leq |Q| \leq |P| - 1$.
- Both P and Q are over the character set $\{$A, B, C, D$\}$.

Sample Input 1

```
3
7 3
ABADCAB
CBB
11 7
ABACCDBADAC
AADCDAC
3 2
ABA
CB
```

Sample Output 1

```
ADCABAB
ABABDCCADAC
NO
```

Problem K
Number with Bachelors

Time limit: 2 seconds

Memory limit: 1024 megabytes

Problem Description

Numbers without duplicated digits are considered bachelor numbers. For example, 123 is a bachelor number in decimal number system, and 9af is a bachelor number in a hexadecimal number system. Both decimal number 101 and hexadecimal number aba are not bachelor numbers since there are duplicated digits in them. In this problem, you get two types of question. For one, given an interval, say, $[a, b]$ in a designated number system, decimal or hexadecimal, you have to figure out the total number of bachelor numbers in the interval, including a and b. For another, you are given a number, say, i in a designated number system you have to find the i^{th} bachelor number in that number system.

Input Format

The first line of the input is a number n, which specifies the number of test cases. Each test case is a question and appears in one line. Each question starts with a letter 'd' or 'h', indicating the question is in decimal domain or hexadecimal domain, respectively. For decimal domain, the following numbers are all represented in decimal. For hexadecimal domain, the following numbers are all represented in hexadecimal. Next to the first letter is a digit 0 or 1, indicating the type of question to be asked. For type 0 question, two integers a and b ($0 \leq a \leq b < 2^{64}$) follow, which represent an interval. For type 1 question, an integer $1 \leq i < 2^{64}$ follows, which represents an order.

Output Format

Output an integer for each question in its corresponding test case. For each question in decimal domain, the answer must be in decimal. For each question in hexadecimal domain, the answer must be in hexadecimal. For type 1 question, if the i^{th} bachelor number does not exist, output a single letter '-' in its corresponding line.

Technical Specification

- $1 \leq n \leq 50000$.
- $0 \leq a \leq b < 2^{64}$.
- $1 \leq i < 2^{64}$.

Sample Input 1

```
6
d 0 10 20
h 0 10 1f
d 1 10
h 1 f
d 1 1000000000
h 1 ffffffffffffffff
```

Sample Output 1

```
10
f
9
e
-
-
```

Problem L
Save lives or money

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

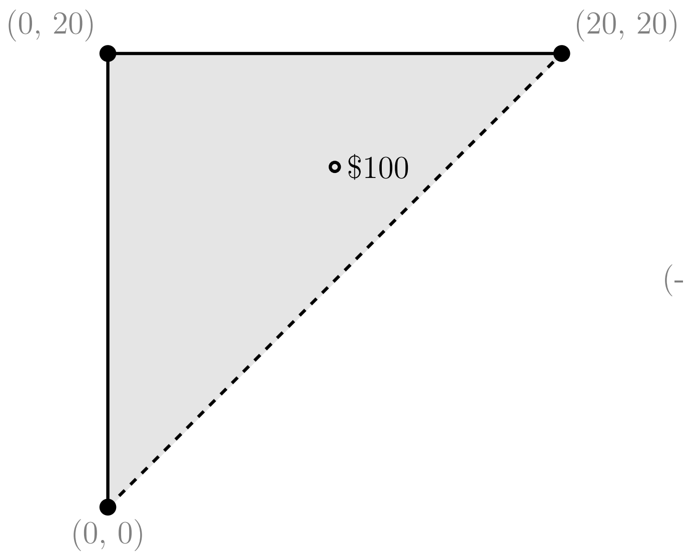
Village "Under The Sea" is located inside a valley. There is a big river in front of the only entry of the village. This year, they encounter a flood that happens roughly once in a century. Because the government is lack of awareness, it is too late to evacuate the residents. The water will flow into the village soon.

Fortunately, this village has walls and gates that could block the water. But we cannot block all the water outside. Otherwise there will be too much water flowing through the river and destroy a nuclear plant in a neighborhood of the village, and brings incalculable damage to everyone. We need to allow some water flowing in, with a manageable way.

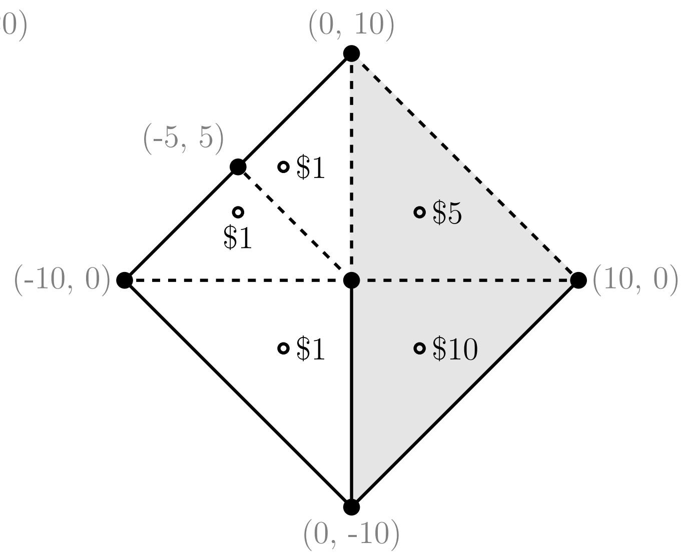
The walls and gates separate the village into many closed regions. Any two different regions could reach each other with exactly one path through the gates if we open all of them. To be clear, the sample 1 is a village consists of 1 region with 2 walls and 1 gate. The solid lines are walls and the dashed line is a gate in the figure below. And the sample 2 is another village consists of 5 regions with 5 walls and 5 gates. Given the estimated water volume, the government could decide to close some gates and leave the rest open. Let the floodwater destroy some regions and leave others safe. The shaded regions in the figures are destroyed regions of the best plans in the sample outputs.

A government official asks you to write a program to help them choosing which gates to open. They give you a list consisted of all the residents with the place they live and money they own. The government official wants you to find a way to save people with the most total wealth. You feel not good to treat rich and poor people differently. So you want to do something different in secret. You will give a plan which save the most people instead. In case there are different plans that save the same number of people, then you choose the one that saves the most money among them.

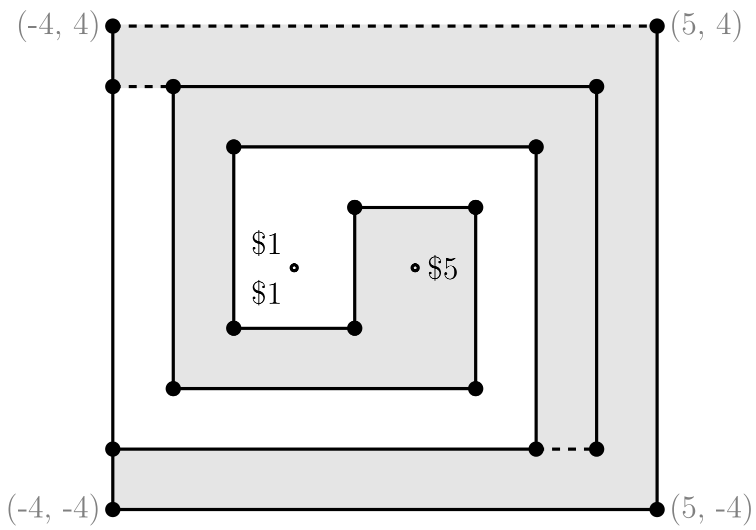
Sample 1



Sample 2



Sample 3



Input Format

The first line contains 1 integer $Area$ – the estimated area that the flood will destroy.

The second line contains 3 integers G, W, and R – the number of the gates, walls, and the residents.

Then G lines follow. Each line contains 4 integers x_{1_g}, y_{1_g}, x_{2_g}, y_{2_g} that represent the coordinates of the two endpoints of a gate.

Then W lines follow. Each line contains 4 integers x_{1_w}, y_{1_w}, x_{2_w}, y_{2_w} that represent the coordinates of the two endpoints of a wall.

Finally, there are R lines. Each line contains 3 integers x_r, y_r, and $money_r$ that represent the coordinates of a resident and the amount of money they owns.

Output Format

You should output 2 lines.

The first line has 1 real number and then 3 integers *area*, *money*, *people*, and *gate_n*, which represent the result of the plan. *area* is a real number rounding to the nearest tenth after the decimal point, which is the total area of destroyed regions. *money* is the total amount of money of the victims. *people* is the number of the victims. *gate_n* is the number of the opened gates.

The second line has *gate_n* integers which are the indices of the opened gates in an arbitrary order. Note that the gates are indexed from 1 to G.

If the *Area* in the input is larger than the village, the *area* you output should be the whole size of the village, the *money* should be the total amount of money of all the people in the village, and the *people* should be all the people in the village. And you should open all the gates.

If the *Area* in the input is no more than the village, the *area* you output should be equal to or larger than *Area*.

If there are multiple solutions that can save the same number of people, choose the one which loses less money. If there are still multiple solutions which lose the same amount of money, choose the one with smaller destroyed area. If there are still multiple solutions which destroy the same size of area, anyone will do.

Technical Specification

- $0 < area, G, W, R < 5000$
- $-5000 < x, y < 5000$
- $0 \leq money < 5000$
- There is exactly one gate on the boundary of the village. The water will flood into the village through this gate. This gate should be opened in a workable plan.
- All the regions are simple polygons. They do not intersect themselves and have no holes.
- All the walls or gates will not intersect with each other. They will touch others only at the endpoints.
- Each endpoint will connect to at least two walls or gates. There is no hanging wall or gate.
- All the positions of the residents will locate in the interior of regions. They will not be outside of the village. And they will not sit right on a wall, a gate, nor a junction.

Sample Input 1

```
20
1 2 1
0 0 20 20
20 20 0 20
0 20 0 0
10 15 100
```

Sample Output 1

```
200.0 100 1 1
1
```

Sample Input 2

```
100
5 5 5
0 10 10 0
0 0 0 10
0 0 10 0
0 0 -10 0
0 0 -5 5
0 -10 -10 0
-10 0 -5 5
0 10 -5 5
10 0 0 -10
0 0 0 -10
3 3 5
-5 3 1
-3 5 1
-3 -3 1
3 -3 10
```

Sample Output 2

```
100.0 15 2 2
1 3
```

Sample Input 3

```
33
3 17 3
-4 4 5 4
-4 3 -3 3
3 -3 4 -3
0 1 0 -1
-4 3 -4 -3
-3 -2 -3 3
-2 2 -2 -1
2 1 2 -2
3 2 3 -3
4 3 4 -3
-3 3 4 3
-2 2 3 2
-2 -1 0 -1
0 1 2 1
-3 -2 2 -2
-4 -3 3 -3
-4 -4 5 -4
-4 -4 -4 -3
-4 3 -4 4
5 -4 5 4
1 0 5
-1 0 1
-1 0 1
```

Sample Output 3

```
48.0 5 1 2
1 3
```

Almost blank page

Problem M
Keystroke

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

You are designing a numeric keypad for numbers 1 to 4, where each number is associated with a unique key. All of the keys are arranged as a 2×2 matrix, and there is a detection circuit beneath the keypad. When a key is pressed, the circuit will transmit the keystroke signals to the controller, which will receive its row number and column number. We can use a pair (`row`, `column`) to represent an event of a keystroke. Precisely speaking, when you press the key of number i where $i \in \{1, 2, 3, 4\}$, the controller will receive the pair $(\lfloor (i-1)/2 \rfloor, (i-1) \bmod 2)$. For example, when you press key 3, the controller gets $(1, 0)$ as the keystroke signal. You would like to press several keys at the same time for some reason. When you do this, the controller can still receive their corresponding row/column numbers. However, their row numbers are mixed together, as well as the column numbers. For example, when you press keys 1 and 4 simultaneously, the controller would get row numbers $\{0, 1\}$ and column numbers $\{0, 1\}$, because key 1 emits $(0, 0)$ and key 4 emits $(1, 1)$. Another example is that when you pressed keys 1 and 2 simultaneously, the controller can only receive $(\{0\}, \{0, 1\})$ because key 1 emits $(0, 0)$ and key 2 emits $(0, 1)$ and their row numbers are the same. Notice that different keystroke combinations may lead to the same signal. Press keys 2 and 3 would get $(\{0, 1\}, \{0, 1\})$ which is identical to press 1 and 4. Press keys $1, 2, 3, 4$ simultaneously would get the same result. Given a keystroke signal, which is in the (`row`, `column`)-paired form, please write a program to identify the total number of possible keystroke combinations that can lead to this signal.

Input Format

The first line of the input is a positive integer that specifies the number of test cases. Each test case follows immediately in the next line after the previous one. In each test case, its first line gives you two positive integers m and n. Its second line gives you m distinct integers that are the received row numbers. Its third line gives you n distinct integers that are the received column numbers. All numbers in the same line are space-delimited.

Output Format

Output the result in a single line for each test case.

Technical Specification

- There are at most 10 test cases.
- $1 \leq m, n \leq 2$.

Sample Input 1

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

Sample Output 1

```
1
1
```

Sample Input 2

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

Sample Output 2

```
7
1
```