

Problem A

Perfect Power

Time limit: 1 second

If a positive integer n can be expressed as the power of another integer greater than 1, then it is called a *perfect power*. For example, both 8 and 9 are perfect powers because $8 = 2^3$ and $9 = 3^2$. Formally, n is a perfect power if there are integers $m > 1$ and $k > 1$ such that

$$n = m^k.$$

Write a program, on input a positive integer n , determine whether n is a perfect power or not. If n is a perfect power, compute m and k so that $n = m^k$, and print them out.

There may be many pairs m and k satisfying the equation $n = m^k$. For example, $625 = 25^2 = 5^4$. If the solution is not unique, print out the solution with largest k .

Input Format

The input file may contain many positive integers, one integer per line. The value of every integer is less than $2^{63} - 1$.

The last line of the input file contains a "0". It indicates the end of the test data.

Output Format

For each integer n in the input file, except the last one (which is a 0), determine whether $n = m^k$ for some positive integers $m > 1$ and $k > 1$. If n is a perfect power, print the values of m and k on one line with a space between them. If n is not a perfect power, then let $m = n$ and $k = 1$, and print them in the same way that n is a perfect power.

Sample Input

```
8
9
12
625
0
```

Output for the Sample Input

2 3
3 2
12 1
5 4

Problem B

Promenade Dance

Time limit: 2 seconds

The NCPC organizing committee holds a promenade dance party for all contestants. Each contestant invites his/her own dance partner. When entering the Ballroom, each contestant picks up a roll of colored ribbon. The contestants then lined up in one single row in order of their contest ID, which are numbered from 1 to n .

The invited dance partners arrive in no particular order. Each person enters the Ballroom and picks up one end of the color ribbon from their invitee and stretch the ribbon as they lined up in a parallel row of dancing partners. When everyone has arrived, there is one row of n contestants and another row of n invited dance partners, with n crisscrossed ribbons that connect contestants and their dance partners. Figure 1(a) shows an example of the initial dance lineup with six contestants and six invited dance partners. In this figure, the invited dance partner of Contestant 1 arrives first, followed by invited dance partner of contestant 6, followed by invited partner of contestant 2, 3, 5 and 4, respectively.

The party host wants to select some pairs for the first dance. The host wants to cut-off the least number of ribbons, so that the remaining pairs with ribbons not crisscrossed are selected for the first dance. In the above example, 4 pairs are selected for the first dance by way of cutting off 2 ribbons (Fig. 1(b)). Note that four pairs is also the maximum that can be achieved in the above example. Given different arrival configurations, determine how many pairs are selected for the first dance.

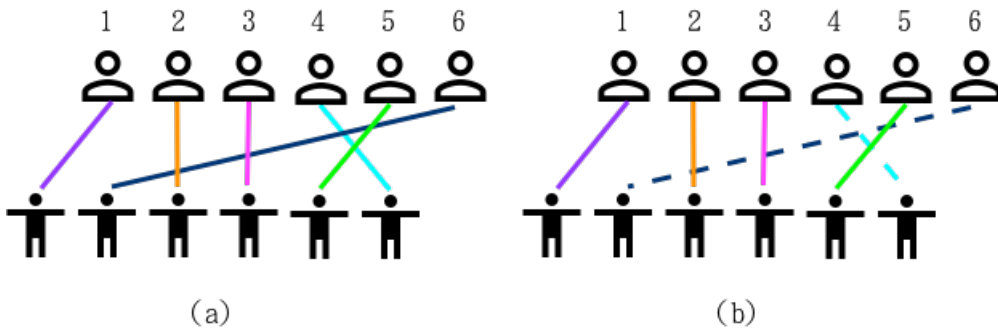


Figure 1: (a) Six contestants with dance partners. (b) Four pairs selected for the first dance.

Input File Format

The first line of input is an integer indicating the number of test cases to follow, which is at most 10. For each test case, the first line contains one integer, n , where $n \leq 20,000$, indicating the number of contestants. The second line contains n distinct integers in the range of 1 to n . The i th number is the Contest ID of the contestant that invited that i th person to the dance.

Output Format

For each test case print out the number of pairs that are selected for the first dance.

Sample Input

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

Output for the Sample Input

```
4
5
```

Problem C

Distribution Distance

Time limit: 3 seconds

Randomized computation is an important technique in Computer Science. However, true randomness is impossible because of the finiteness of the currently available computational models. What we can have is pseudo-randomness. Bob is interested in designing pseudo-random number generator, which outputs the so-called “random numbers” ranging from 1 to N . Recently, he designs two methods to generate the pseudo-random numbers. He wants to know how close are these two methods, and experiments for M times. Let $c_{i,j}$ be the number of times that the integer j appears in the experiments with method i , for $i = 1, 2$ and $j = 1, \dots, N$. That is, $\langle c_{1,1}, \dots, c_{1,N} \rangle$ is the statistics of method 1, and $\langle c_{2,1}, \dots, c_{2,N} \rangle$ is for method 2. Note $\sum_{j=1}^N c_{i,j} = M$, for $i = 1, 2$. Let $[N]$ denote the set $\{1, \dots, N\}$. Given the two statistics, we want to determine the value

$$\max_{A \subseteq [N]} \left| \sum_{j \in A} c_{1,j} - \sum_{j \in A} c_{2,j} \right|,$$

which is the so-called *distribution distance*.

For example, let $N = 6, M = 10$ and two sequences be $\langle 2, 1, 1, 3, 0, 3 \rangle$ and $\langle 1, 0, 1, 4, 1, 3 \rangle$. Then the answer is 2 for $A = \{1, 2\}$ or $\{1, 2, 6\}$. Note that it is possible to have more than one subset to achieve the maximum value. Your task is to write a program to help Bob find the distance between the two statistics.

Technical Specification

1. $1 < N < 100$
2. $1 < M < 10,000$

Input File Format

The first line of the input gives the number of test cases, T (< 10). Each test case has three lines. For each case, the first line consists of two positive integers N, M indicating the range of pseudo-random number and the number of experiments made, respectively. Each of the next two lines contains N non-negative integers: $c_{i,1} c_{i,2} \dots c_{i,N}$, separated by space(s) and with the sum M , indicating the statistics of experiments.

Output Format

For each test case, output one line that contains the value of distribution distance.

Sample Input

```
2
6 10
2 1 1 3 0 3
1 0 1 4 1 3
10 7
1 1 1 0 0 1 1 1 1 0
0 0 0 4 2 0 1 0 0 0
```

Sample Output for the Sample Input

```
2
6
```

Problem D

Identifying Normal Samples

Time limit: 5 seconds

Ben is analyzing a collection of n tumor samples S_1, S_2, \dots, S_n . Each tumor sample is represented by a set of mutations. All samples are nonempty and distinct. Ben suspects that some of the samples are corrupted. Based on his knowledge in biology, Ben believes that some of the samples, called *normal samples*, are more reliable and worth further investigation. Let us say that a pair of two distinct samples (S_i, S_j) is a *conflicting pair* if $S_i \not\subset S_j$ and $S_j \not\subset S_i$ (i.e., neither of them is a subset of the other). A sample S_k is *normal* if there does not exist any conflicting pair (S_i, S_j) such that $S_k \subset S_i$ and $S_k \subset S_j$. For example, let $n = 6, S_1 = \{a, b, c\}, S_2 = \{a, b, d\}, S_3 = \{b, c\}, S_4 = \{c\}, S_5 = \{a, b\}$, and $S_6 = \{a\}$. The sample S_4 is contained in S_1 and S_3 . Since (S_1, S_3) is not a conflicting pair, S_4 is normal. The sample S_6 is contained in S_1, S_2 , and S_5 . Since (S_1, S_2) is a conflicting pair, S_6 is not normal. Similarly, the set S_5 is not normal because it is contained in S_1 and S_2 and (S_1, S_2) is a conflicting pair. Ben wants to find all normal samples. Please write a program to help him.

In this problem, the input is an $n \times n$ matrix M which represents the relations between any two samples. More specifically, for $1 \leq i \neq j \leq n$, $M_{i,j} = 1$ if $S_i \subset S_j$, $M_{i,j} = 2$ if $S_i \supset S_j$, and $M_{i,j} = 3$ if (S_i, S_j) is a conflicting pair. For $1 \leq i \leq n$, $M_{i,i} = 0$. For the example mentioned above, the matrix M is illustrated as follows.

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	0	3	2	2	2	2
S_2	3	0	3	3	2	2
S_3	1	3	0	2	3	3
S_4	1	3	1	0	3	3
S_5	1	1	3	3	0	2
S_6	1	1	3	3	1	0

Technical Specification

1. $1 \leq n \leq 4000$.
2. M is consistent with a set of distinct nonempty samples S_1, S_2, \dots, S_n .

Input File Format

The first line of the input file contains an integer indicating the number of test cases. Each test case begins with a line containing an integer n . Each of the next n lines gives a row of M : the j^{th} character of the i^{th} line represents the entry $M_{i,j}$.

Output Format

For each test case, print a line containing the number of normal samples.

Sample Input

```
3
6
032222
303322
130233
131033
113302
113310
7
0223322
1021133
1101133
3220133
322033
1333303
1333330
1
0
```

Output for the Sample Input

```
4
5
1
```


Problem E

Archery at the Summer Olymlic Camp

Time limit: 3 seconds

The Summer Olymlic Camp (SoCamp) is about to start, and you are asked to design a computer program to help judge the archery sport!

The regions on the target are represented by convex polygons in the plane and are denoted by C_1, C_2, \dots, C_n . It is guaranteed that, for any $1 \leq i < j \leq n$, the polygon C_i contains C_j in its interior, i.e., the regions form a nesting sequence. In addition, each region, say C_i , is associated with a score p_i .

The evaluation of a shot is simple. If the arrow falls inside a region, say, C_i , or falls on its boundary, the player gets p_i points (from C_i). Otherwise, it is considered irrelevant (to C_i). Hence, the score of a shot is the total scores the shot gets from all the regions.

Given the regions, C_1, C_2, \dots, C_n , the points p_1, p_2, \dots, p_n they are worth, and a sequence of shots $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$, your task is to output the score of the shots.

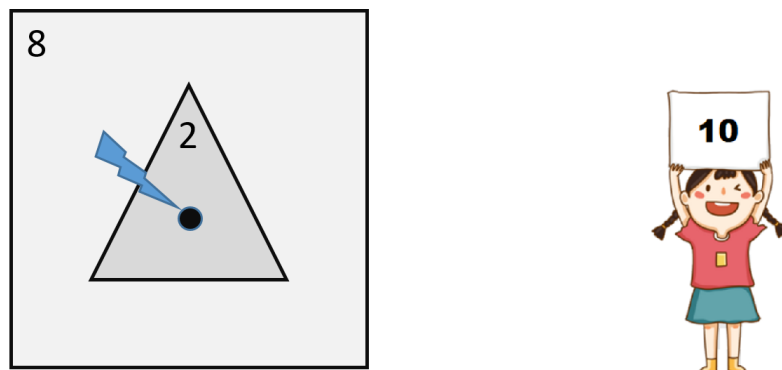


Figure 1: In this example, there are two convex polygons C_1 with $p_1 = 8$ and vertices $\{(-2, 2), (-2, -2), (2, -2), (2, 2)\}$ and C_2 with $p_2 = 2$ and vertices $\{(0, 1), (-1, -1), (1, -1)\}$. With shot located at $(0, 0)$, the player gets 10 points from the two polygons.

Input Format

There are multiple test cases in the input file, separated by a blank line.

Each test case starts with two integers n and m in a line, which are the number of regions and the number of shots. Then it comes n regions in order. Each region C_i starts with two integers p_i and k_i in a line, which are the score of C_i and the number of vertices of C_i . Then there are k_i lines, one for each vertex of C_i in counter-clockwise (CCW) order and

each containing two integers which are the coordinates of the vertices. After the n regions, there are m lines, one for each shot and each of which contains two integers that are the coordinates of the shots.

A test case with $n = m = 0$ indicates the end of the input.

You may additionally assume the following.

- $0 \leq n, m \leq 10^5$, $0 \leq p_i \leq 10^3$, $3 \leq k_i \leq 10^5$.
- The absolute value of all coordinates does not exceed 10^9 .
- The total number of vertices in the input does not exceed 10^6 .

Output Format

For each test case, print the score of each shot in a line.

Sample Input

```
2 3
8 4
-2 2
-2 -2
2 -2
2 2
2 3
0 1
-1 -1
1 -1
0 0
3 4
-2 1

0 0
```

Output for the Sample Input

```
10
0
8
```

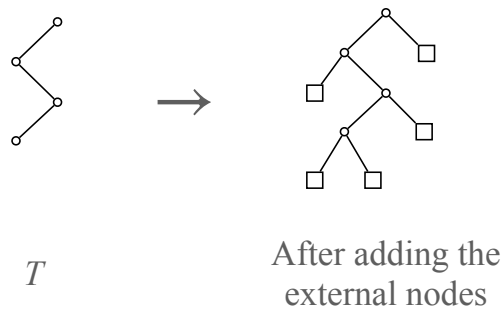
Problem F

The Secrete Map

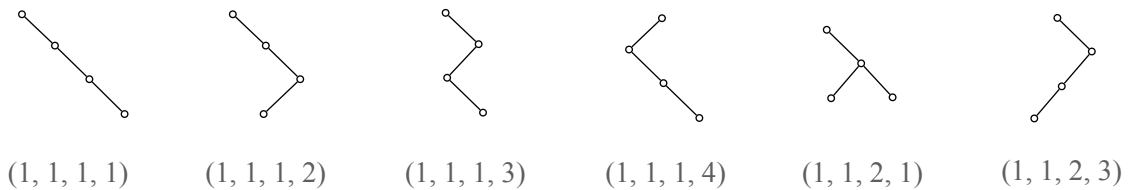
Time limit: 1 second

A treasure hunter H has a secrete message, with which he knows that the treasure is hidden in a cave. There are branches in the cave, so it is impossible to reach the treasure without a map. Fortunately, from the secrete message he knows that the routes in the cave form a **binary tree**, with the entrance being the root. However, there is no tree structure given in the message, but two numbers n and r . He asks a computer scientist Professor W for help. After a few days of investigation, Professor W tells H that n is the number of nodes in the binary tree, and r is the order of the tree in a specific ordering. The order r is the lexicographic order of an encoding of the n -node binary tree. The encoding of a binary T is defined as follows. First, add children, called **external nodes**, to all the nodes so that the root has degree 2 and every non-root node has degree 3. For each node of T , mark it by the number of external nodes in its **left subtree**. The encoding of T is an integer sequence resulting from an **inorder traversal** of T .

For example, there is a 4-node binary tree given in the left of the following figure. After adding the external nodes, we have the tree on the right side. Traversing T using inorder traversal, we have a sequence (1, 1, 2, 4). This is the 7th 4-node binary tree, so in the secrete message the two given numbers are 4 and 7.



The first six 4-node binary trees are given in the following figure.



Given n and r , write a program to find the encoding of the corresponding tree.

Input File Format

The first line is a positive integer N , $N \leq 10$, which is the number of test cases. For each test case, the input has only one line, containing two positive integers n and r , with $1 \leq n \leq 15$ and $r \leq 10^6$. It is guaranteed that r is less than or equal to the number of n -node binary trees. The two numbers are separated by a space.

Output Format

N lines, which are the outputs of the test cases in the given order. For each test cases, the output has one line, containing n integers, which is the encoding of the r th n -node binary tree. Consecutive integers are separated by a space.

Sample Input

```
2
4 7
5 42
```

Output for the Sample Input

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

Problem G

Assigning Frequencies

Time limit: 1 second

The NCPC kingdom has n cities, numbered $0, 1, \dots, n-1$. There are also some bidirectional roads, each connecting two distinct cities. Sensibly, each road intersects only with the two cities that it connects. Because the engineers do not know how to build viaducts, the roads in the kingdom lie flat on the ground, without any two roads crossing each other. To avoid traffic jam, furthermore, each city is incident to at most four roads.

To facilitate wireless communication, the NCPC queen wants to assign each city a frequency for emitting radios. The frequencies available are 0, 1, 2 and 3. To avoid interference, any two cities connected by a road must receive different frequencies. Please help the queen.

Input File Format

The first line is the number (≤ 10) of test cases. Each test case is as follows: The first line is the number $n \leq 10000$ of cities. The second line is the number m of roads. Then each of the next m lines contains a road, specified by the two cities that it connects. Two consecutive numbers in a line are separated by a space.

Output Format

For each test case, output the frequencies of cities $0, 1, \dots, n-1$, in that order. If there are many correct outputs, just output one of them. It is guaranteed that a solution exists.

Sample Input

```
2
6
11
1 5
2 1
0 1
3 1
4 3
4 5
0 5
3 0
```

2 4
2 5
4 0
5
9
0 1
4 0
4 1
3 4
2 4
0 2
1 2
3 1
2 3

Output for the Sample Input

0 1 0 2 1 2
0 1 2 0 3

Problem H

Managing a Tree Network

Time limit: 4 seconds

Paul, the manager of a computer network, encounters a problem as follows. The network managed by Paul is an undirected tree T , in which each vertex v has a positive integer weight $w(v)$ and each edge (u, v) has a positive integer length $l(u, v)$. Paul selects an integer bound b and wants to decompose the tree T into subtrees, each of which will be managed by a local administrator, according to the following conditions:

1. The subtrees are obtained by removing edges of T .
2. The total weight of any subtree is at least b .

To make a local administrator manage a subtree more simply, Paul hopes to have as many subtrees as possible. Paul also hopes that the total length of the removed edges should not be too large. More specifically, Paul wishes to maximize the number of subtrees; and in case there are several ways to maximize the number of subtrees, he wants the way that minimizes the total length of the removed edges.

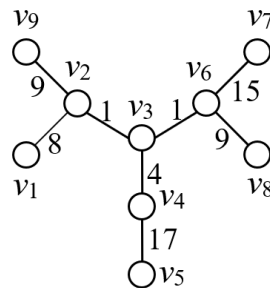


Figure 1: A tree T , in which $(w(v_1), w(v_2), \dots, w(v_9)) = (6, 3, 3, 8, 1, 2, 7, 5, 5)$.

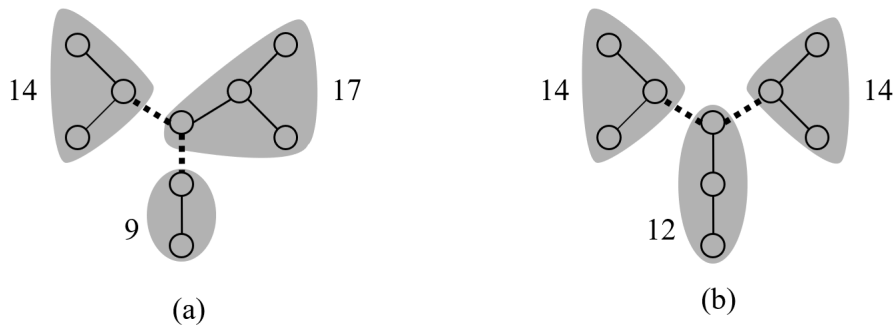


Figure 2: Two feasible decompositions, in which $b = 9$.

Consider the tree T in Figure 1. Let $b = 9$. Two feasible ways to decompose T are given in Figure 2. In Figure 2(a), there are three subtrees, whose total weights are 14, 9, and 17 respectively; and the total length of the removed edges is $1 + 4 = 5$. In Figure 2(b), there are also three subtrees, whose total weights are 14, 14, and 12 respectively; and the total length of the removed edges is $1 + 1 = 2$.

Your task is to help Paul to compute the maximum number of subtrees along with the minimum total length of the removed edges.

Technical Specification

1. The number, n , of vertices of T is an integer between 1 and 10^4 .
2. The bound, b , is an integer between 1 and 100.
3. The weight of each vertex is an integer between 1 and 10^3 .
4. The length of each edge is an integer between 1 and 10^3 .

Input File Format

The first line of the input file contains an integer indicating the number of test cases. Each test case begins with a line containing two integers n and b , where $1 \leq n \leq 10^4$ and $1 \leq b \leq 100$. The integer n indicates the number of vertices of T and the integer b is the minimum total weight of a subtree. Then, n lines follow, each containing an integer $w_i, 1 \leq w_i \leq 1000$, indicating the weight of vertex $v_i, 1 \leq i \leq n$. Then, $n - 1$ lines follow, each containing three integers $i, j, l, 1 \leq i, j \leq n, 1 \leq l \leq 1000$, indicating there is an edge of length l connecting vertices v_i and v_j .

Output Format

For each test case, print a line containing two integers k and l^* . The integer k indicates the maximum number of subtrees; and l^* indicates, when T is decomposed to k subtrees, the minimum total length of the removed edges. If there is no feasible way to decompose the tree T , print '-1' instead of two integers k and l^* .

Sample Input

```
4
9 9
6
```


3
3
8
1
2
7
5
5
1 2 8
9 2 9
6 3 1
3 2 1
4 3 4
5 4 17
6 8 9
6 7 15
7 9
7
5
3
6
8
8
1
1 2 9
2 4 2
4 6 8
7 5 5
5 3 1
3 1 4
1 2
1
1 1
1

Output for the Sample Input

3 2
3 3
-1
1 0

Problem I

Tao

Time limit: 2 seconds

Tao is the path to truth. It is often symbolized as a *Tai Chi diagram* in the traditional Chinese culture, and it generates two modes: *Yin* and *Yang*. These two modes keep in flux, and they circulate into four forms: major Yin, minor Yin, major Yang and minor Yang, respectively. These four forms generate the *eight trigrams*. This idea can be used to explain the fundamental principles of reality.

One day, you invented a mysterious quantum machine that can grasp the *encoding* of Tao for anything. When you feed something into your awesome machine, it outputs a string that records the past, current and even the future of this entity. However, useful information is usually hidden; you need to do some *mining*. The first step you want to do is to identify *fundamental elements* in this string. Following the aforementioned philosophical thought of Tao, you realize that a fundamental element is a substring that is symmetric and contains all distinct letters in this string. A fundamental element is called a *tao* if it cannot be reduced by removing any characters in either end. That is, a tao cannot contain any other tao as a substring. For example, consider the string `aaabaabb`. It has several substrings that are symmetric, e.g. `aaa`, `aba`, `aabaa`, and `baab`. The first substring `aaa` is not a fundamental element because it has no `b`. The second `aba` and the fourth `baab` are taos. The third `aabaa` is a fundamental element but not a tao because it contains `aba`, a tao, as a substring. Please write a program to find all taos in a given string.

Input File Format

The first line gives you a positive integer n which is the total number of test cases, where $1 \leq n \leq 20$. Each test case has a single string that occupies one line with at most 10^5 characters. The characters in a string can only be lowercase English letters, but the set of distinct letters may be different in different strings. Notice that this implies that a substring may be a tao in one string but not a tao in another one because a tao only needs to contain all distinct letters in the resident string.

Output Format

For each test case, please find all of the occurrences of taos in the given string. Then, output the total number and total length of all of the occurrences in one line, where two integers are separated by a single space. The total length is calculated by adding the lengths of all substrings that are taos for this test case.

Sample Input

5
abaaba
aaabaabb
abcd
aaaaa
abcddcbaabcddcba

Output for the Sample Input

3 10
2 7
0 0
5 5
3 24

Problem J

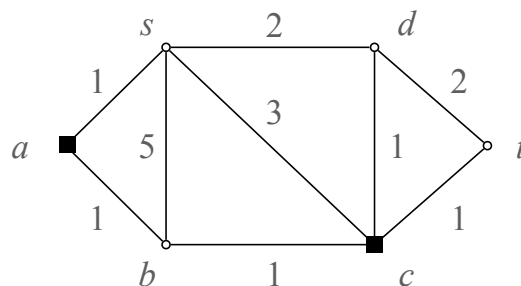
Shortest Paths with Unvisited Sightseeing Spots

Time limit: 2 seconds

During the summer vacation, you plan to visit NCPC national park. There are many sightseeing spots in the park. On the day you leave, you want to take photos of some unvisited spots. However, you do not have enough time so you decide to have the photos along a shortest route from the hotel s to the exit t . Given the map of NCPC national park, which is a graph, the position of the hotel, the position of the exit, and a list of unvisited vertices, please write a program to find the maximum number of photos you can take.

You can assume that the graph is undirected, simple, and connected. Each edge of the graph has a positive weight, representing the time needed to travel between the two spots at the endpoints. In addition, neither s nor t is in the list of unvisited vertices.

For example, the graph in the following figure has four shortest s, t -paths, $s \rightarrow d \rightarrow t$, $s \rightarrow d \rightarrow c \rightarrow t$, $s \rightarrow c \rightarrow t$, and $s \rightarrow a \rightarrow b \rightarrow c \rightarrow t$. Unvisited sightseeing spots are marked by squares. The last shortest path contains two unvisited sightseeing spots, which is more than those in the other shortest s, t -paths. Thus, the requested number is 2.



Input File Format

The first line of the input is a positive integer N , $N \leq 16$, which is the number of test cases. For each test case, the first line contains 5 integers, n , e , s , t , and p , where

- n is the number of vertices of the graph, with $1 \leq n \leq 10^5$. The vertices are numbered from 0 to $n - 1$.
- e is the number of edges of the graph, with $1 \leq e \leq 10^5$.
- s is the vertex number of the hotel.
- t is the vertex number of the exit, $t \neq s$.
- p is the number of unvisited sightseeing spots, with $1 \leq p \leq n - 2$.

Each of the following e lines contains 3 integers, u , v , and w , which specifies the weight w of the edge with endpoints u and v , where $1 \leq w \leq n$. The last line of the input contains p integers, $0 \leq p \leq n$, which are the unvisited sightseeing spots. Consecutive numbers in a line are separated by a space.

Output Format

N lines, which are the outputs of the test cases in the given order. Each line is a nonnegative integer, which is the maximum number of unvisited vertices along a shortest s, t -path of the test case.

Sample Input

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

Output for the Sample Input

```
2
0
```

Problem K

Overlapping Triangles

Time limit: 3 seconds

Given two triangles, what is the overlapping area? A triangle is represented by three points (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) . For example, two triangles are represented as $(1, 1)$, $(1, 2)$, $(2, 1)$, and $(1, 1)$, $(1, 2)$, $(2, 2)$. The overlapping area forms a triangle $(1, 1)$, $(1, 2)$, $(3/2, 3/2)$ with area $1/4$. Be careful that degenerate triangles are also taken as a valid one, such as $(1, 1)$, $(1, 1)$, $(1, 2)$.

Technical Specification

1. $1 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 32$
2. $1 \leq T \leq 80$

Input Format

The first line contains an integer $T(\leq 80)$ representing the number of test cases. For each test case, there are 2 triangles A and B . The coordinates of triangles A and B are listed orderly in a line. For each triangle, there are 6 integers x_1, y_1, x_2, y_2, x_3 , and y_3 .

Output Format

For each test case, please output the overlapping area ans . If ans is an integer, please output the integer value. Otherwise, please output ans as the irreducible fraction n/d . Be careful that during the calculation of n or d , some value may be greater than 2^{64} .

Sample Input

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

Sample Output for the Sample Input

0
1/4
0
1/2

Problem L

Majority Vote

Time limit: 1 second

The Pacific island of Baxos was a thriving mercantile center. Wealth led to political sophistication, and the Baxons replaced their tribal theocracy with a parliamentary form of government. But trade came before civic duty, and no one in Baxos was willing to devote his life to Parliament. As a result, the Baxon Parliament had to function even though legislators continually wandered in and out of the parliamentary Chamber. So, they needed to elect legislators through a unique voting system.

Suppose there're N legitimate voters on the island and there're K candidates for the Parliament. Every legitimate citizen simply ranks the candidates with a *unimodal permutation* on $[K] := \{1, 2, \dots, K\}$. A permutation $\pi = (\pi_1, \dots, \pi_K)$ is called unimodal if there is an integer $i \in \{1, \dots, K\}$ satisfying $\pi_1 > \dots > \pi_i$ and $\pi_i < \pi_{i+1} < \dots < \pi_K$. Note π_i indicates the rank of candidate i , the smaller the better. Given the N individual rankings on the candidates, the Parliament wants to determine a permutation $\tau = (\tau_1, \dots, \tau_K)$ as the preference order, such that for any pair of i and j , $1 \leq i < j \leq K$, the *majority* of the N voters prefer candidate τ_i to candidate π_j . Note the *majority* means at least half of the voters prefer candidate τ_i to candidate π_j .

Ranking	Candidate 1	Candidate 2	Candidate 3
Voter 1	1	2	3
Voter 2	3	1	2
Voter 3	3	2	1

For example, let $N = 3$, $K = 3$ and the voters give three rankings: $(1, 2, 3)$, $(3, 1, 2)$, $(3, 2, 1)$, as illustrated in the above table. Then the answer is $\tau = (2, 3, 1)$. It can be verified the majority of the voters (2 voters) favors Candidate 2 over the other candidates. Also, the majority prefers Candidate 3 to Candidate 1. Your task is to write a program to help the parliamentary Chamber find the preference order as the group decision.

Technical Specification

1. $1 < N < 1,000$
2. $1 < K < 1,000$

Input File Format

The first line of the input gives the number of test cases, T (< 10). For each test case, the first line consists of two positive integers N, K indicating the number of voters and the number of candidates, respectively. The i -th line of the following N lines contains K integers indicating the ranking given by the i -th voter.

Output Format

For each test case, output one line that contains the preference order of the candidates.

Sample Input

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

Sample Output for the Sample Input

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