Árboles y Grafos, 2024-2

Para entregar el sábado 23 de noviembre de 2024
A las 23:59 en la arena de programación

Instrucciones para la entrega

- Esta tarea es de carácter opcional y completamente individual. No se puede usar en su desarrollo ningún tipo de ayuda de otras personas, de internet o de inteligencia artificial. No se deben buscar los problemas en plataformas de internet, soluciones al problema en internet ni casos de prueba. No es posible tener conversaciones o reuniones con compañeros del curso ni con otras personas. Cualquiera de estas acciones se entenderá como una falta a las reglas de la tarea y al código de honor del curso.
- Para esta tarea y todas las tareas futuras en la arena de programación, la entrega de soluciones es *individual*. Por favor escriba claramente su nombre, código de estudiante y sección en cada archivo de código (a modo de comentario). Adicionalmente, cite cualquier fuente de información que utilizó. Los códigos fuente que suba a la arena de programación deben der de su completa autoría.
- En cada problema debe leer los datos de entrada de la forma en la que se indica en el enunciado y debe imprimir los resultados con el formato allí indicado. No debe agregar mensajes ni agregar o eliminar datos en el proceso de lectura. La omisión de esta indicación puede generar que su programa no sea aceptado en la arena de programación.
- Puede resolver los ejercicios en C/C++ y Python.
- Debe enviar sus soluciones a través de la arena. Antes de subir sus soluciones asegurese de realizar pruebas con los casos de pruebas proporcionados para verificar que el programa finalice y no se quede en un ciclo infinito.
- El primer criteríо de evaluación será la aceptación del problema en la arena cumpliendo los requisitos indicados en los enunciados de los ejercicios y en este documento. El segundo criterio de evaluación será la complejidad computacional de la solución y el uso de los temas vistos en clase. **Es necesario incluir en la cabecera del archivo comentarios que expliquen la complejidad de la solución del problema para cada caso**.

  En adición a lo anterior, para efectos de la calificación se tendrán en cuenta aspectos de estilo como no usar `break` ni `continue` y que las funciones deben tener únicamente una instrucción `return` que debe estar en la última línea.
- La solución de cada uno de los problemas permitirá obtener una bonificación de hasta **0.7** décimas. El total acumulado se sumará en las notas de las demás tareas realizadas durante el curso.

Problemas prácticos

Hay cinco problemas prácticos cuyos enunciados aparecen a partir de la siguiente página.

A - Problem A

Source file name: `almost.cpp`
Time limit: x seconds

I hope you know the beautiful Union-Find structure. In this problem, you're to implement something similar, but not identical.

The data structure you need to write is also a collection of disjoint sets, supporting 3 operations:

1 p q**:** Union the sets containing p and q. If p and q are already in the same set, ignore this command.

2 p q**:** Move p to the set containing q. If p and q are already in the same set, ignore this command.

3 p**:** Return the number of elements and the sum of elements in the set containing p.

Initially, the collection contains n sets: $\{1\}$, $\{2\}$, $\{3\}$, . . ., $\{n\}$.

Consider the following example with $n = 5$ sets:

- Initially: $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$.
- After operation 1 1 2: $\{1, 2\}, \{3\}, \{4\}, \{5\}$.
- After operation 2 3 4: $\{1, 2\}, \{3, 4\}, \{5\}$ (the empty set that is produced when taking out 3 from $\{3\}$ is omitted).
- After operation 1 3 5: $\{1, 2\}, \{3, 4, 5\}$.
- After operation 2 4 1: $\{1, 2, 4\}, \{3, 5\}$.

Input

There are several test cases. Each test case begins with a line containing two integers n and m ($1 \leq n, m \leq 100\,000$), the number of integers, and the number of commands. Each of the next m lines contains a command. For every operation, assume $1 \leq p, q \leq n$.

The input must be read from standard input.

Output

For each type-3 command, output 2 integers: the number of elements and the sum of elements.

The output must be written to standard output.

Sample Input	Sample Output

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

B - Problem B

Source file name: `friends.cpp`
Time limit: x seconds

These days, you can do all sorts of things online. For example, you can use various websites to make virtual friends. For some people, growing their social network (their friends, their friends' friends, their friends' friends' friends, and so on), has become an addictive hobby. Just as some people collect stamps, other people collect virtual friends.

Your task is to observe the interactions on such a website and keep track of the size of each person's network.

Assume that every friendship is mutual. If Fred is Barney's friend, then Barney is also Fred's friend.

Input

The first line of input contains one integer specifying the number of test cases to follow. Each test case begins with a line containing an integer F, the number of friendships formed, which is no more than 100000. Each of the following F lines contains the names of two people who have just become friends, separated by a space. A name is a string of 1 to 20 letters (uppercase or lowercase).

The input must be read from standard input.

Output

Whenever a friendship is formed, print a line containing one integer, the number of people in the social network of the two people who have just become friends.

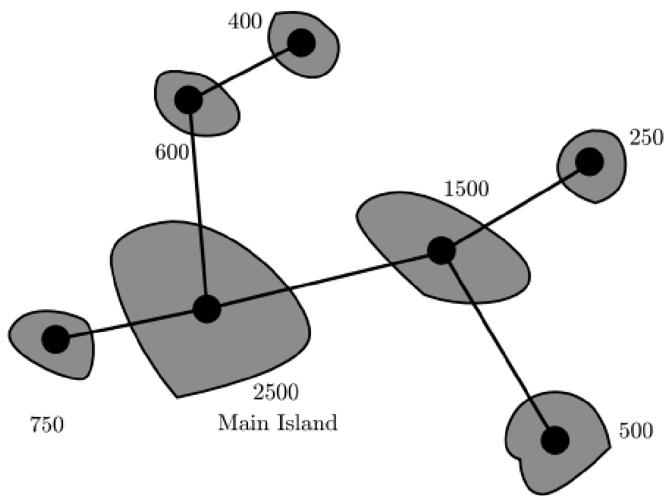
The output must be written to standard output.

Sample Input	Sample Output
1 3 Fred Barney Barney Betty Betty Wilma	2 3 4

C - Problem C

Source file name: `island.cpp`
Time limit: x seconds

The company Pacific Island Net (PIN) has identified several small island groups in the Pacific that do not have a fast internet connection. PIN plans to tap this potential market by offering internet service to the island inhabitants. Each groups of islands already has a deep-sea cable that connects the main island to the closest internet hub on the mainland (be it America, Australia or Asia). All that remains to be done is to connect the islands in a group to each other. You must write a program to help them determine a connection procedure.



For each island, you are given the position of its router and the number of island inhabitants. In the figure, the dark dots are the routers and the numbers are the numbers of inhabitants. PIN will build connections between pairs of routers such that every router has a path to the main island. PIN has decided to build the network such that the total amount of cable used is minimal. Under this restriction, there may be several optimal networks. However, it does not matter to PIN which of the optimal networks is built.

PIN is interested in the average time required for new customers to access the internet, based on the assumption that construction on all cable links in the network begins at the same time. Cable links can be constructed at a rate of one kilometer of cable per day. As a result, shorter cable links are completed before the longer links. An island will have internet access as soon as there is a path from the island to the main island along completed cable links. If m_i is the number of inhabitants of the i^{th} island and t_i is the time when the island is connected to the internet, then the average connection time is:

$$\frac{\sum t_i * m_i}{\sum m_i}$$

Input

The input consists of several descriptions of groups of islands. The first line of each description contains a single positive integer n, the number of islands in the group ($n \leq 50$). Each of the next n lines has three integers x_i, y_i, m_i, giving the position of the router (x_i, y_i) and number of inhabitants m_i ($m_i > 0$) of the islands. Coordinates are measured in kilometers. The first island in this sequence is the main island.

The input is terminated by the number zero on a line by itself.

The input must be read from standard input.

Output

For each group of islands in the input, output the sequence number of the group and the average number of days until the inhabitants are connected to the internet. The number of days should have two digits to the right of the decimal point. Use the output format in the sample given below.

Place a blank line after the output of each test case.

The output must be written to standard output.

Sample Input	Sample Output
7 11 12 2500 14 17 1500 9 9 750 7 15 600 19 16 500 8 18 400 15 21 250 0	Island Group: 1 Average 3.20

D - Problem D

Source file name: `permutation.cpp`
Time limit: x seconds

Given N and K find the N-th permutation of the integers from 1 to K when those permutations are lexicographically ordered. N starts from 0. Since N is very large N will be represented by a sequence of K non-negative integers $S_1, S_2, \ldots, S_k$. From this sequence of integers N can be calculated with the following expression.

$$\sum_{i=1}^{K} S_i \cdot (K-i)!$$

Input

First line of the input contains T (≤ 10) the number of test cases. Each of these test cases consists of 2 lines. First line contains a integer K ($1 \leq K \leq 50000$). Next line contains K integers $S_1, S_2, \ldots, S_k$. ($0 \leq S_i \leq K - i$).

The input must be read from standard input.

Output

For each test case output contains N-th permutation of the integers from 1 to K. These K integers should be separated by a single space.

The output must be written to standard output.

Sample Input	Sample Output
4 3 2 1 0 3 1 0 0 4 2 1 1 0 4 1 2 1 0	3 2 1 2 1 3 3 2 4 1 2 4 3 1

E - Problem E

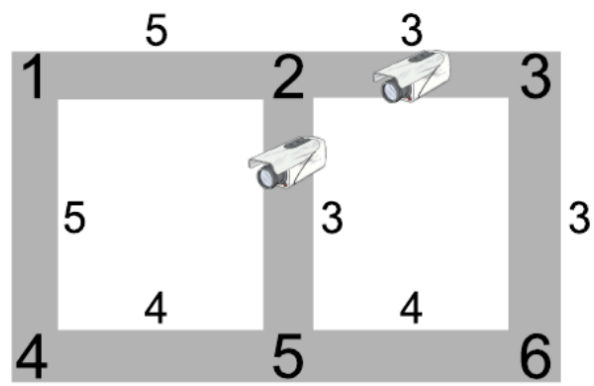
Source file name: `racing.cpp`
Time limit: x seconds

Singapore will host a Formula One race in 2008. The race will be held on a 5.067km long street circuit, consisting of 14 left hand turns and 10 right hand turns. In the run up to the F1 race, the number of illegal night street racing activities have been on the rise. Such races consists of several rounds around a designated street circuit.

The authorities would like to deploy a new vehicle monitoring system in order to catch these illegal Saint Andrew's Road, part of the Formula One circuit racers. The system consists of a (Kenny Pek, Piccom) number of cameras mounted along various roads. For the system to be effective, there should be at least one camera along each of the possible circuits.

The Singapore road system can be represented as a series of junctions and connecting bidirectional roads. A possible racing circuit consists of a start junction followed by a path consisting of three or more roads that eventually leads back to the start junction. Each road in a racing circuit can be traversed only in one direction, and only once.

Your task is to write a program that computes the optimal placement of the vehicle-monitoring cameras. You will be provided with a description of a connected road network to be monitored in terms of the roads and junctions. The junctions are identified by the bigger numbers in the figure below. A camera can be deployed on the roads (and not the junctions).



The cost of deploying a camera depends on the road on which it is placed. The smaller numbers by the roads in the figure indicate the cost of deploying a camera on that road. Your job is to select a set of roads that minimizes the total cost of deployment while ensuring that there is at least one camera along every possible racing circuit (i.e., loop in the road network).

Input

The input consists of a line containing the number c of datasets, followed by c datasets, followed by a line containing the number '`0`'.

The first line of each dataset contains two positive integers, n and m, separated by a blank, which represent the number of junctions and number of roads, respectively. You may assume that $0 < n < 10\,000$ and $0 < m < 100\,000$. For simplicity, each one of the n junctions are labeled from 1 to n. The following m lines of each dataset each describes one road. Each line consists of three positive integers which are the labels of two different junctions and the cost of deploying a camera on this road. The cost of deploying a camera is between 1 and 1 000.

The input must be read from standard input.

Output

The output consists of one line for each dataset. The c-th line contains one single number, representing the minimal cost of setting up the vehicle monitoring system such that there is at least one camera along every possible circuit.

The output must be written to standard output.

Sample Input	**Sample Output**
1 6 7 1 2 5 2 3 3 1 4 5 4 5 4 5 6 4 6 3 3 5 2 3 0	6