Fifth Annual Juilfs Contest

Spring 2013

DOCTOR WHICH

A. Transmat (5 points)

The Fourth Great and Bountiful Human Empire is built on transmat technology. Two terminals are created using quantum entanglement. Then, one of the terminals is placed on a spaceship, travelling at 80% of the speed of light, and placed on a distant planet. Matter placed on one terminal is then instantly transported to the other terminal. However, the original transmat devices connected many planets to each other. Since the terminals have to be continuously powered through the network, the power consumption from the reactor in London would be enormous, and they found that the resulting network cannot be powered. The Imperial Parliament has decided to build a bigger reactor, but first eliminate most of the devices. They want everyone to be able to travel to all parts of the empire, but breaking it up into several steps so that the power drain on the new reactor is minimal.

They would like you to figure out power usage after all the obsolete terminals are decommisioned. Since the power usage is determined by the space between terminals, you should leave enough terminals to traverse the entire Empire, but the distance over the entire network should be minimized. That is, the sum of distances between all terminals should be as low as possible.

As input, you will be given a line containing a single integer $0 < n < 100$ where n is the number of problems to follow. Each problem begins with a line containing a single integer $0 < m < 1000$ where m is the number of pairs of transmat terminals. Each pair of transmat terminals is listed on a single line. The two planet names are listed first, followed by the floating point distance in astronomical units. The distance will be not more than 158099308 (The furthest planet in the Empire). Since planets change positions, the distance is the maximum distance between the two planets. Your output should consist of the total distance of a minimum network. This will inform the Imperial Corps of Engineers how large to make the reactor.

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td><pre>2
4
Earth Jupiter 6.45
Earth Mars 2.52
Earth Moon 0.0026
Moon Mars 2.5226
14
Earth Mercury 1.48
Earth Venus 1.73
Earth Mars 2.66
Earth Europa 6.4500045
Earth Ganymede 6.4500071
Europa Ganymede 0.0000124
Europa Callisto 0.0000178
Ganymede Callisto 0.0000230
Earth Titan 11.1100024
Earth Rhea 11.1100126
Titan Rhea 0.0000123
Earth Triton 31.4435475
Triton Oberon 50.6258352
Earth Oberon 21.0658352</pre></td><td><pre>8.9726
75.9394321</pre></td></tr>
</table>

B. Weeping Angel (3 points)

The touch of a Weeping Angel will send a person back in time with the Angel gaining sustenance from the change in potential energy. Each angel has a period. For example, one Angel sends its victim back 53 years. Another sends its victim back 48 years. A group of angels gang up on the Doctor's companion and send her back in time but all touching her at once. This sets up a chronol-harmonic resonance that causes the companion to repeatedly travel back in time in a sequence that ends at the beginning of the Universe. Each step in the sequence is a multiple of all the angel's periods. If the doctor can use the TARDIS to travel back in time to the first step in the sequence, he can stop the effect. For example, if the two angels above touched someone simultaneously, the person would travel back in time 2544 years; materialize; travel back in time another 2544 years; materialize again; travel back 2544 years, etc until the person reaches the beginning of the universe. The Doctor can only stop the process if he travels to the first materialization 2544 years before the touch.

As input, you will be given a line with a single number $0 < n \leq 100$ where n is the number of lines that follow. This will be followed with n lines. Each line will have a space-delimited list of integer angel periods in years. There will be no more than twenty numbers on each line. For each line, you should output the numbers of years that the companion will be sent back in time for the first materialization. If you miscalculate the first materialization, the companion will be sent back to the beginning of the universe and be annihilated in the Big Bang.

Sample Input	Sample Output

`4`
`2 3 5`
`10 20 40`
`15 3 13 15 21 5`
`101 103`
	`30`
	`40`
	`1365`
	`10403`

C. Pieces of Cake (4 points)

"Well, that was a piece of cake, eh K-9?" "Piece of cake, Master? Radial slice of baked confection...coefficient of relevance to Key of Time: zero." Dr. Who has a bunch of friends who want to split up a cake, and it looks like K-9 isn't going to help us divide it up, so it's up to you, noble programmer.

Suppose that there are *n* people sharing a circular cake with *k* flavors. However, these players all prefer various flavors of cake more than others! We want to split up the cake so that everyone gets something that they believe is fair.

How do we determine if a portion of the cake is fair? Each player has something called a "preference ratio." From this, we create a system of points for each player and use it to assign a value to the entire cake, and then use it to assign a value to a portion. If the value of the portion is at least 1/n of the value of the cake, the player will view it as fair. We'll illustrate this with an example:

Suppose that there are two flavors of cake, chocolate and vanilla, and that 120 degrees of the cake are chocolate, and 240 degrees of the cake are vanilla. Let a player prefer chocolate to vanilla in a ratio of 3:2. Then for every degree of chocolate, we'll assign 3 points, and for every degree of vanilla, we'll assign 2 points. This player will evaluate the cake at $3\times120 + 2\times240 = 840$ points. If there are two players, this player will feel that any portion having at least $840 / 2 = 420$ points is fair. A portion of 100 degrees of chocolate and 100 degrees of vanilla will have a value of $100\times3 + 100\times2 = 500 > 420$, and therefore be fair. However, a portion of 120 degrees of chocolate and 20 degrees of vanilla will have a value of $120\times3 + 20\times2 = 400 < 420$, and is therefore unfair.

It's your job to divide up the cake and give everyone portions that they believe are fair. The players here are not altruistic. They are content with other players receiving portions that they believe are too small, as long as they get what they want. Also, assume that you want no leftovers, that is, all parts of the cake should be given to someone.

INPUT:

The first line will be a single integer, T, the number of trials. This will be followed by T problems. Each problem will have a single line with two integers N and K, separated by whitespace. These represent the number of players and flavors, respectively.

The next line will contain K whitespace-separated integers. These represent the degrees that each flavor has on the cake.

After this line will be N lines of K whitespace-separated integers, which represent the preference ratio of each player.

continued on reverse

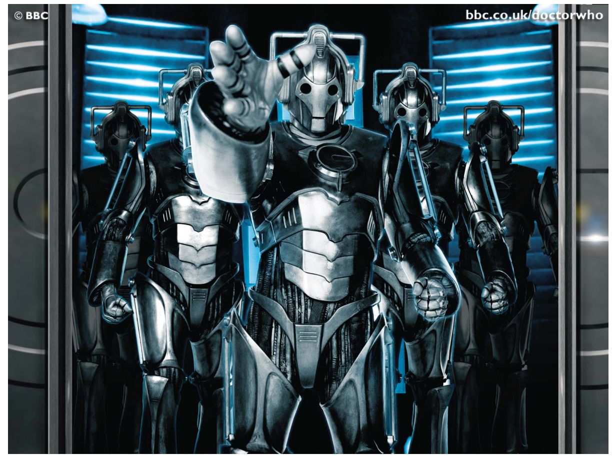
OUTPUT:

For each problem, output N lines. These N lines begin with "PLAYER i" in ascending order, starting from 1. After this, output K numbers, which represent the K flavors from the input. These numbers can be either integers or floats. The output to each problem should be separated by a blank line.

Note: The output is not unique, and any division of the cake that all players deem fair will be considered correct. This is one possible solution, and you are encouraged to find another solution to the sample input.

Sample Input	Sample Output
2 2 2 120 240 3 2 1 4 3 4 20 50 80 210 1 2 5 1 2 6 1 2 10 2 3 1	PLAYER 1: 100 100 PLAYER 2: 20 140 PLAYER 1: 0 0 80 70 PLAYER 2: 0 50 0 70 PLAYER 3: 20 0 0 70

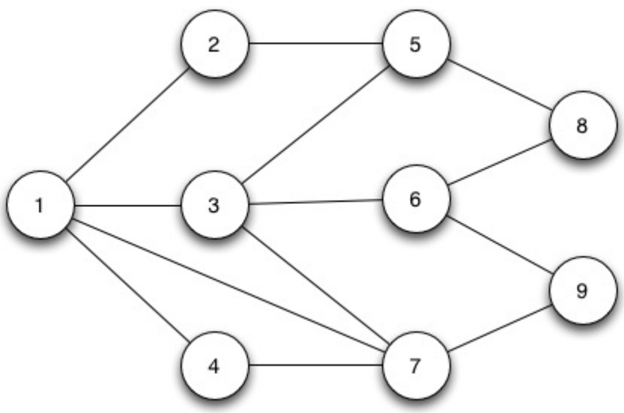
D. Find Possible Allies (3 points)



In a dangerous world full of Cybermen, Dr. Who is looking for possible friends with whom he can form alliances against his enemies. He has been able to determine who among the Cybermen are his enemies, but he is not sure whom he could work with as an ally. He has decided that he will try to contact those among the Cyberman who are enemies of his enemies and see if an alliance is possible. Of course, such a contact had better not be known to already be an enemy of Dr. Who.

More precisely, the enemy list is described as a graph G. If (A,B) is an edge of the graph, then A and B are enemies (and so (B,A) is also an edge). We say that A and C are possible allies if there is a node B so that (A,B) and (B,C) are both edges of G. But also, we need to require that (A,C) is not an edge of G.

Nodes are numbered 1 to N. Dr. Who is designated as node number 1, and the Cybermen are nodes numbered between 2 and N. The desired output will be all nodes C where there is a path of length two from node 1 to node C and there is not a path of length one from 1 to C. So in the attached sample graph the possible allies are nodes 5, 6, and 9 (it is acceptable to provide node 1 as a potential ally as well).



INPUT:
In the input the first line will be an integer `numCases`, the number of input cases. Each case will start with two lines, the first an integer N, the number of nodes, and M, the number of edges. There will then follow M lines, each line containing two integers A and B, indicating that A and B are enemies. In other words, both (A,B) and (B,A) are edges of the graph G.

Assumptions:

- Each A and B are between 1 and N.
- A could be larger or smaller than B, but A will not equal B. (No one can be an enemy of themself.)
- Each input case G represents a connected graph (there are no isolated nodes).
- We do not consider paths consisting of three or more edges - they have no effect as to who might be an ally.

continued on reverse

OUTPUT:

A list of the nodes at distance two from node number 1 that are not at distance one.

Sample Input	Sample Output

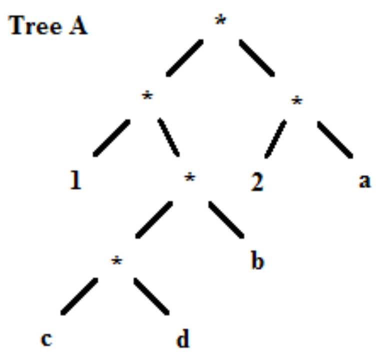
`2`
`9`
`13`
`1 2`
`1 3`
`1 4`
`1 7`
`2 5`
`3 5`
`3 6`
`3 7`
`4 7`
`5 8`
`6 8`
`6 9`
`7 9`
`4`
`3`
`2 1`
`2 3`
`4 3`
	`1 5 6 9`
	`1 3`

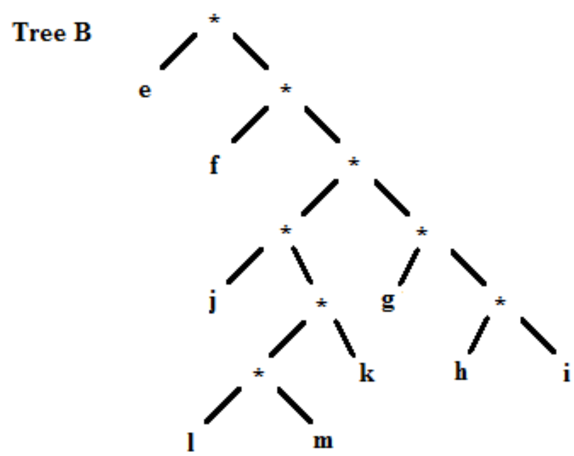
E. Decoding the Daleks (4 points)

The Daleks have developed a coded form of communication that their leaders use to distribute orders with the rest of the Daleks. However, in their perpetual rage, they did not devise a very clever encryption algorithm. Instead, they use a Huffman encoding tree. Sometimes, they send a false signal that does not have a valid key. The Doctor needs you to help him determine which keys are valid so he can know what they are up to.

The key is a representation of a Huffman encoding tree as a breadth-first search. The key is created by listing each node in the tree, starting with the root, and printing each depth-level of nodes as the tree grows. Left-children are listed before right children. Remember that a Huffman encoding tree is binary, but not necessarily balanced. Nodes without values are listed as an asterisk (*) – every asterisk (*) has two non-null children. Numbers and letters are end-nodes in the tree.

For example:





Tree A's key is: ***1*2a*bcd

Tree B's key is: *e*f***j*g**khilm

Input:

Input will consist of a single number, n>0, indicating the number of keys to follow, followed by n lines of input, each with a new key. Each key consists of numbers, letters, and the * character. No key will have duplicate characters (other than the * character).

Output:

For each key, you will print YES if the key represents a valid binary tree, or NO if it does not. Print each response to a new line.

continued on reverse

Sample Input	Sample Output
5 *12 12 ***ab8e ***abe *b*c*3***124k	YES NO YES NO YES

F. Tardis Time-Space Travel on a Hyper-cylinder (5 points)

The TARDIS, which is the Doctor's vehicle of choice, moves along Time And Relative Dimensions In Space. It is bigger on the inside, but jettisons interior space in return for energy when out of fuel. The Doctor needs to know how many square feet will be lost, in order to decide where and when to go next. He has the cylindrical-temporal co-ordinates of some time-and-space locations to which he can potentially escape.

These co-ordinates are written in terms of the TARDIS's current location and orientation, which is (0, 0, 0, 0).

- r = Horizontal Distance [0 to ∞);
- a = Horizontal Angle [0 to 2π);
- z = Distance Up ($-\infty$ to ∞)
- t = Temporal Distance ($-\infty$ to ∞)

Your task is to find, from the given co-ordinates, the closest one to which to escape. If the two closest co-ordinates to (0,0,0,0) are the same distance, then choose the first one. In addition, there are several cases, so you will need to answer this question several times.

As a reminder, in case your TARDIS-programmer's manual has worn out from over-use, here are the relevant formulae:

- Distance from (r_1, a_1, z_1, t_1) to (r_2, a_2, z_2, t_2) = sqrt[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 + (t_2 - t_1)^2]
- Conversion from cylindrical-temporal to rectangular-temporal co-ordinates: (x, y, z, t) = (r * cos(a), r * sin(a), z, t)

Input:

The first line of input is a number numCases$\geq$1, the number of cases. For each case, the first line `N` is the number of co-ordinates. Each co-ordinate is a line starting with a single character (the "name" of the co-ordinate, and four real numbers `r a z t`.

Output:

For each case, give the name of the first closest co-ordinate on a separate line. Thus, there will be N lines of output, each a single character.

Sample Input	Sample Output
2 3 A 3 2.17 6 7 B 1.5 0 0 2	B Z

C 144 1.44 14 41 4 Z 0 0 0 0 G 1 1 1 1 i’m just making stuff up here	