Delfts Kampioenschap Programmeren
Problemset

October 4, 2008

A Lanterns

The parents of Hansel and Gretel would like to take their children out on a walk by night through the forest. Family hiking trips are a cherished family tradition, but Hansel and Gretel do not walk as fast as their parents. During previous walks, they therefore often lost track of them, causing them to be left on their own. At night the forest is a scary place, with predators lurking everywhere. Therefore, it is not hard to understand the children are not particularly looking forward to this walk.

Hansel and Gretel, who do not have GPS facilities, decide to prepare themselves for the walk by bringing lanterns to the forest by day. They can attach these lanterns to trees in the forest. At night, each lantern will then provide a circular spot of light (shadow effects are not taken into account). As predators do not like light, Hansel and Gretel will not be devoured as long as they will stay within the light provided by the lanterns. If they will attach lanterns to trees in the forest that are situated close enough to each other, the corresponding light spots will touch or overlap. This way, a light zone encompassing multiple trees is created, in which they can walk safely. First of all, Hansel and Gretel wonder how many trees can at most be illuminated in one connected light zone (the 'largest connected light zone' in terms of illuminated trees). If the largest connected light zone consists of multiple trees, there will only be one zone in the forest with this maximal number of trees.

Of course Hansel and Gretel can attach lanterns to all trees in this zone, but if trees are situated really close to each other, the light spot of one tree can actually already illuminate the other tree. Therefore, Hansel and Gretel might be able to bring less lanterns without reducing the number of trees they can see and reach in the lighted zone. As this reduces the weight to carry ánd the amounts of light pollution and emitted CO_2, they now also are very interested in the smallest number of lanterns needed to illuminate all trees in the largest connected light zone, while maintaining the possibility to reach any tree in this zone from any other tree without having to step into the dark (see Figure 1 for an illustration). Poor Hansel and Gretel do not just lack GPS facilities, they also don't have a computer in their log cabin. You do, so maybe you can help them.

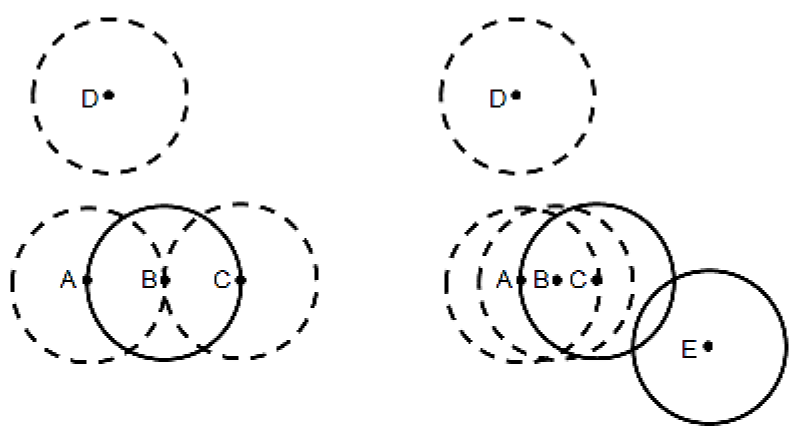

Figure 1: Two possible tree situations. On the left, at most three trees can be illuminated in one light zone. Just one lantern is necessary to achieve this: it should be attached to tree B. On the right, at most four trees can be illuminated in one light zone; the most efficient way to do is is attaching lanterns to trees C and E. In both cases, tree D is situated too far from the other trees to be able to form a light zone with them.

A.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 100$), the number of test cases;
- Then, for each test case:
  - A line with an integer number r ($0 < r \leq 100$), which is the radius of the circular light spot that the lanterns create when illuminated;
  - A line with an integer number t ($1 < t \leq 20$), describing the number of trees in the forest;
  - Then, for each tree:
    - A line with the tree location, expressed in an (x, y) integer coordinate ($-10000 \leq x \leq 10000$, $-10000 \leq y \leq 10000$). Within one test case, each tree will have a unique location.

A.2 Output

For each test case, the output contains one line with two integer numbers, separated by a space. The first number indicates the size of the largest connected light zone (in terms of illuminated trees) that can be created, the second number indicates the minimal amount of lanterns needed to illuminate all these trees.

A.3 Example

Input	Output
2	3 1
2	4 2
4	
0 0	
2 0	
-2 0	
0 6	
2	
5	
0 0	
1 0	
-1 0	
0 6	
4 -2	

Table 1: Example input and output

B Burger King

After the programming contest team MSX decides to grab a bite at the Burger King. It turns out it's a rather busy evening there: each counter has a queue of people waiting to order their meals and more people keep pouring in. Since they're rather hungry from a long day of programming, team MSX wants to order food as soon as possible.

Team member Menno suggests that, in order to get their food as fast as possible, they should join the shortest queue. Team member Sarah disagrees: she notices that some employees are considerably slower than their colleagues, hence the team should join the queue of the fastest employee. Team member Xenon, however, notices that every now and then, some employees switch with their colleagues. If a customer was being helped during the change, this takes time, since the new employee has to start all over again in helping the customer. If the customer's order would have been finished at the moment of the change, though, the customer's order is finished first and the new employee immediately continues serving the next customer without delay. Besides that, the service speed is also dependent on the type of customer: a family takes considerably more time to help than a middle-aged man.

Team MSX decides it's best to work dynamically. All team members have a strong predictive ability, which allows them to know how much time an employee will need to help a certain customer. As they enter the Burger King, they calculate for each queue hoe long it will take until all customers in that queue have been helped. They then join the queue that will be the first one to be empty.

When the situation changes, for example when the employee at a counter changes, they reevaluate which queue they should join to be able to order as soon as possible. If it turns out it's more effective to join another queue, they will not hesitate. If team MSX has to choose from multiple queues that would be equally effective, they choose the queue with the lowest identification number, unless they're in one of those queues themselves, in which case they will not switch queues. Of course, when the team switches queues, they join the end of the new queue. The switching itself takes no time at all.

Given a set of employees, queues of customers and a schedule of events, figure out how many minutes it takes before the team can order their well-earned diner if they choose their queues optimally.

B.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 10000$), the number of test cases;
- Then, for each test case:
  - A line with one positive integer m ($1 \leq m \leq 10$): the number of queues;
  - For each queue two lines:
    - A line with three positive integer numbers i ($0 \leq i < m$): the (identification-) number of the queue, i_c ($0 < i_c \leq 30$): the number of customers in the queue at the moment that team MSX enters the Burger King, and e_c ($0 \leq e_c \leq 10$): the minimum amount of minutes that the employee for queue i takes to help a customer. The numbers are separated by a space;
    - A line with i_c positive integer numbers, separated by spaces, where each number represents a customer in queue i and the *value* ($0 \leq value \leq 15$) represents the amount of extra minutes it takes for an employee to help this customer. The first customer in the list is the one that is standing in front of the employee (and will immediately be helped by the employee) while the last one is at the end of the queue.
  - A line with one positive integer v ($0 \leq v \leq 100$): the number of events taking place;
  - Then for each event a line containing one of the following:

* For the event of a new customer joining a queue: The string "join" and three positive integer numbers t_v ($0 < t_v \leq 300$), the moment (beginning of the minute) at which the event occurs, i_v ($0 \leq i_v < m$), the queue at which a new customer joins and c_v ($0 \leq c_v \leq 15$), the amount of extra minutes it takes for an employee to help this customer. The string and numbers are separated by spaces. A new customer always takes place last in line;
* For the event of an employee being replaced by another: the string "change" and three positive integer numbers t_v ($1 \leq t_v \leq 300$), the moment (beginning of the minute) at which the event occurs, i_v ($0 \leq i_v < m$), the queue at which the employee will be replaced by a new one and e_v ($0 \leq e_v \leq 10$), the minimum amount of minutes that the new employee for queue i_v takes to help a customer. The string and numbers are separated by spaces.

No more than one event takes place in the same minute. In addition, there will never be more than 30 customers in a queue (excluding team MSX) at any time.

B.2 Output

For each test case the output contains one line with a single integer: the number of minutes team MSX needs to wait before they can order their food.

B.3 Example

Input	Output
1 2 0 4 2 4 2 6 2 1 6 3 1 2 1 1 3 1 6 join 1 1 3 join 4 1 4 join 6 0 10 join 8 1 8 change 2 1 2 change 5 0 4	34

Table 2: Example input and output

C Labyrinth

A well-known hero, let's say his name is Conan, is on a quest. On his way to quest completion, Conan finds himself at the entrance of a labyrinth. There is no way around the labyrinth, so Conan will have to enter and find the exit at the other side.

Because Conan knows very little about the labyrinth he is about to enter and also lacks psychic abilities, at every intersection he randomly chooses how to continue on his search for the exit. Conan does have a perfect memory and will make sure that he will never enter a path in the same direction twice. Moreover, he only goes back on a path where he came from when he is absolutely sure that this will take him closer to the exit. So, whenever he's exploring a path, he will explore that path as far as he can, before going back.

Your task is to write a program that, given a map of the labyrinth, determines the expected number of steps Conan will take through the labyrinth. The exit of the labyrinth is always reachable from the entrance. There will be no cycles or empty rooms (of dimension 2x2 or above) in the labyrinth. A passage next to the exit may still be considered an intersection.

```
##s###
#....#
#t####
```

Conan enters the labyrinth on the position marked "s". The exit of the labyrinth is marked "t". There will be exactly one entrance and exactly one exit.

In the above example, Conan will do one step to the south. Now he has to decide whether to continue eastward or to the west. If he goes to the east, he will return after four steps to his position. If he goes to the west, he will exit the labyrinth in two steps.

The expected number of steps is:

$$1 + \frac{1}{2} \cdot (4 + 2) + \frac{1}{2} \cdot 2 = 5$$

C.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 100$), the number of test cases;
- Then, for each test case:
  - One line with two positive integers: $3 \leq h, w \leq 96$, the size of the labyrinth;
  - h lines with w characters each, with the following meaning:
    - A character `#` is a wall;
    - The characters `s` and `t` denote the entrance and exit respectively;
    - A character `.` is a passage.
  - The "border" of the labyrinth will be a wall (all characters `#`), except for the entrance and exit, which will always be on this border.

C.2 Output

For each test case, the output contains one line with one number: the expected number of steps from start to exit of the labyrinth, rounded to exactly two decimal places.

C.3 Example

Input	Output

1
3 6
`##s###`
`#....#`
`#t####`
	5.00

Table 3: Example input and output

D Walking the dog

You happen to own a dog, a labrador. Because your pet tends to get lazy and fat without enough exercise, you often take your buddy out for a walk in the local park.

As you already get enough exercise yourself and don't feel like running after your dog, you have bought a very long rope to tie your labrador to. Assume it has infinite length. You find yourself a nice place to sit down, holding one end of the rope, and let your dog run free (while still attached to the other end of the rope). While your pet runs through the park the rope could wind around the flagpole that is standing in the middle of the park. You let this happen, but in order not to get entangled, you occasionally jump over the line when it passes the place where you sit.

Your job is, given the points to where the dog runs successively, to determine how many entire times the rope has been wrapped around the flagpole when the dog has finished its walk (so round your answer down to an integer). You may assume that your dog always takes the shortest path to the next point. When this path goes right through the flagpole, the dog always goes around it counterclockwise. For this problem we lay a coordinate map ($-10^9 \leq x, y \leq 10^9$) over this park. The flagpole is at point $(0, 0)$.

D.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 10000$), the number of test cases;
- Then, for each test case:
  - A line with an integer number m ($0 < m \leq 1000$), the number of points the dog's walk consists of;
  - Then, m lines with two integer numbers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9, (x_i, y_i) \neq (0, 0)$), the coordinates of the i'th point the dog will run to.

D.2 Output

For each test case, the output contains one line with one integer number: the number of entire windings the rope has made around the flagpole after you sit down at (x_1, y_1) and let your dog make the walk $(x_1, y_1) \rightarrow (x_2, y_2) \rightarrow \ldots \rightarrow (x_m, y_m)$.

D.3 Example

Input	Output
1 6 0 4 3 -1 -4 -3 -1 -4 -1 2 3 3	1

Table 4: Example input and output

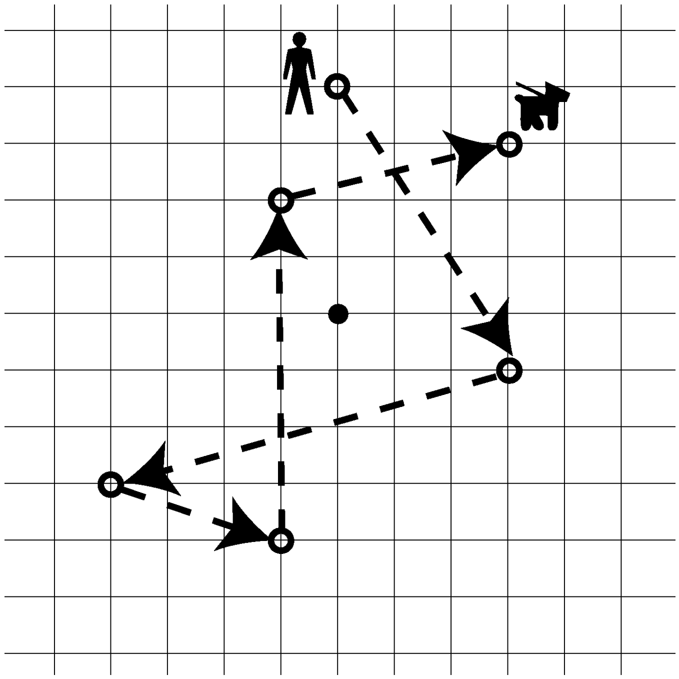

Figure 2: The dog's walk in the example.

E Power cables to sewer pipes

In a municipality somewhere in the country a couple of villages lie close together. In one of the villages there is a large windmill park, that provides the electricity for all villages in the municipality. To transport all this energy, a network of high-tension cables has been constructed that connects the cities to the windmill park, either directly or indirectly (via another village). To prevent power outage when some of the cables break, some redundant high-tension cables have also been added between some of the villages. Altogether, the electricity network in the municipality is in a very good state.

However, due to rust and bad maintenance, the condition of the sewer system of the municipality is a lot worse. That's why the mayor has decided all sewer pipes have to be replaced. There's only one problem: because of extreme shortage, not enough materials are available to construct new sewer pipes. In an attempt to solve this problem, the mayor takes a peculiar decision: all redundant cables in the power network will be removed (putting up with possible power losses) and will be recycled to sewer pipes. As long as all villages will remain connected to each other by power cables this will not give any trouble. Although the villages have done a good job building redundant power cables, they've not been wasting any. They're sure they won't find more than 400 kilometers of redundant cable.

The cables will be recycled by the local metalworks. Over there, they know that 1 kilometer of power cable will yield 1 meter of sewer pipe. The metal workers also know that their machines are only capable of constructing sewer pipes with a length that is an integer multiple of 1 meter. They haven't decided yet how many pipes of each length they are going to produce, so they give a manager the task to make a list of all options, so they can present it to the mayor. For example, if 3 kilometers of power cable can be removed, one option is to produce one sewer pipe of length 3 meters. Another option is to produce one pipe of 2 meters and one pipe of 1 meter long. The third and last option is to produce 3 pipes that are 1 meter long.

Your goal is to determine how many options are going to be on that list. You may assume that, since a lot of sewer pipe is needed, all available redundant power cables are recycled.

E.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 10000$), the number of test cases;
- Then, for each test case:
  - A line with one integer number m ($2 \leq m \leq 100$), the number of villages;
  - A line with one integer number k ($1 \leq k \leq 1000$), the number of power cables;
  - k lines with three integers f_i, t_i, l_i, separated by spaces, indicating there is a power cable between village f_i ($1 \leq f_i \leq m$) and village t_i ($1 \leq t_i \leq m$), which has length l_i ($1 \leq l_i \leq 400$) kilometers

E.2 Output

For each test case, the output contains one line with one number: the number of combinations of sewer pipes the metalworks can make by recycling as much power cable as can be removed from the power network without disconnecting a village from it. If the power network turns out not to be redundant so no cables can be removed, output the number 0.

E.3 Example

Input	Output
1 4 4 1 2 4 1 3 3 1 4 2 3 4 5	7

Table 5: Example input and output

F Sascha

Chinese people are known for their difficulty with pronouncing the “R”. Often the “R” is replaced by the “L”, like in “radio”, which becomes “ladio”. Especially when a secondary language is involved, the examples become very colourful. For example, in Spanish the “V” is pronounced like a soft “B”. In South-America it is most disturbing to hear a Dutch young couple shout “*F*amos, *F*amos!”. To give another well-known example, the English pronunciation of “th” is in The Netherlands often replaced by a loud “S”, which results, for example, in announcing the program “Nor*S* and Sou*S*”.

At a young age, children often have difficulties with certain letters as well (although luckily, they usually get over it quite soon). Sascha, a 3 year old girl, also doesn´t succeed in pronouncing a number of letters. She just replaces them with another letter which she finds easier to pronounce. It became a problem when she started to replace many letters by the same letter. This forced her parents to use a lot of creativity and imagination when trying to figure out what the original word was that she meant. Additional skills were required when they noticed that her “replacement”-rules weren’t consistent. For example, Sascha can clearly pronounce an “R” at the beginning of a word, while substituting the “R” in the middle of a word. This can even happen within the same word!

Your job is to write a translation program to figure out the original word Sascha wanted to say. You have to find the most likely word, given a dictionary of proper words. The most likely word is the word in the dictionary that can be found with the least number of substitutions. A substitution consists of replacing a single letter by another letter. The different letters which cause speech difficulties are not given. Neither is the replacement letter.

F.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 10000$), the number of test cases;
- Then, for each test case:
  - One line with the word as Sascha pronounced it;
  - One line with an integer w ($0 < w \leq 10000$), the number of words in the dictionary;
  - Then w lines with one word each: the dictionary.

All words consist of lower case letters and never exceed 128 characters in length. All words in the dictionary have the same length as the word that Sascha pronounced.

F.2 Output

For each test case, the output contains one line with one word: the word that Sascha most likely meant to say. When multiple words would be possible, a word that appears earlier in the dictionary is more likely.

F.3 Example

Input	Output
1 lolly 5 child mommy hello drink horse	mommy

Table 6: Example input and output

G Choir

Maud is the choirmaster of a choir, and she is preparing a performance of some four-part songs. Four-part means that not every singer sings the same notes, the composer wrote four different parts, and the parts are divided among the singers. The highest part is called the Soprano part, the second highest part is the Alto part, which is followed by the Tenor part, while the lowest part is the Bass part. These different parts are usually denoted (abbreviated) S, A, T, B.

In a perfect world, every singer always sings the part that fits him or her best: women with a high voice (sopranos) sing the soprano part, women with a low voice (altos) sing the alto-part, and the same goes for the male voices (tenor and bass). Of course reality is different. In a way reality is better than perfect. Some singers are not strictly bass or tenor, but are able to sing both tenor and bass parts (not at the same time, though). This is quite helpful because genuine tenors are rare. In the same way some women are able to sing both alto and tenor parts. These multi-talents are rare, however. In Maud's choir only four women are able to sing both alto and tenor, and only three men are able to sing both tenor and bass.

For every song, Maud has decided which singer sings which part. Because of a lack of tenors, some singers sing alto in one song, and tenor in another. Others sometimes sing tenor, and sometimes bass. In this choir sopranos always sing the soprano part. The choir is quite small, so they sing standing in a single line (actually half a circle). Singers singing the same part are standing together: at the leftmost (seen from the choirmaster) the singers singing the soprano part, then the singers singing the alto part, the tenor part and at the rightmost hand the singers singing the bass part. Because singers sometimes sing a different voice in a different song, they have to change places between the songs. This is disturbing, and should be avoided as much as possible. Here is the problem: order the songs in such a way that the number of replacements during the program is minimal. If between two songs two singers have to change place, it counts as two replacements. If the singer at position 1 moves to position 5, four other singers on position 2 upto 4 have to shift, making up for a total of 5 replacements.

G.1 Input

- The first line of input consists of the integer number n $(0 < n \leq 100)$, the number of test cases;
- Then, for each test case:
  - A line containing two positive integers $m \leq 6$ (the number of songs) and $s \leq 20$ (the number of singers);
  - m lines describe for each song the division of the parts over the singers. For simplicity the singers are numbered 1 to s. The division is then denoted as a string of length s. The string BBSAT indicates that singer one sings the bass part, as does singer two. Singer three sings the soprano-part, and so on. Each song has exactly four parts.

G.2 Output

For each test case your program should print the total number of replacements during the performance if the program order and the placement of the singers is chosen such as to minimize the number of replacements. Follow the format of the sample.

G.3 Example

Input	Output
2 2 12 SSSAATTTBBBB SSSAAATTTBBB 3 8 SATBSATB SABTSTAB SABTSATB	0 4

Table 7: Example input and output

H Recycling proteins

Proteins are large organic substances made of amino acids, arranged in a linear chain and joined together by peptide bonds. The amino acids the chain consists of and the order in which they appear determine the physical and chemical properties of the protein. There are 20 standard types of amino acids, each one having a unique 3-letter code, like for example 'Ala' for Alanine. Proteins can be found anywhere, and play an important part inside the human body. Some well-known proteins are hemoglobin, keratin, collagen and insulin.

Two notable scientists have made a breakthrough discovery by designing a machine that utilizes certain undisclosed chemical processes to alter chains of amino acids, thereby effectively converting some protein into another. This enables them to transform organic waste into useful substances like insulin.

The only problem is that the chemical processes for dissolving and creating the peptide bonds used inside the machine are quite expensive. Even a small amount of the protein being recycled contains a lot of amino acid chains, and each of those needs to be processed separately. Your job is to write a program to tell the scientists how much it costs to transform one chain of amino acids into another.

The protein-recyclers give you some details about the costs: removing one instance of an amino acid molecule from a chain costs 2 euro. Inserting one amino acid molecule costs 4 euro. Because of special details in the chemical processes, removing one molecule and directly adding another in its place (effectively replacing one molecule with another) only costs 5 euro. You may assume that the scientists will have sufficient supplies of each of the amino acids to be added. The amino acids Lysine (Lys), Valine (Val), Arginine (Arg) and Histidine (His) are quite rare however, and therefore more expensive. Adding them or replacing an amino acid with one of these 'rare' types costs one euro more than the normal operation.

H.1 Input

- The first line of input consists of the integer number n ($0 < n \leq 10000$), the number of test cases;
- Then, for each test case:
  - A line with an integer number k ($1 \leq k \leq 1000$), the number of amino acid molecules in the protein chain we want to recycle;
  - k lines with a 3-letter string x_i, the type of the ith amino acid molecule in the protein chain we want to recycle;
  - A line with an integer number m ($1 \leq m \leq 1000$), the number of amino acid molecules in the protein chain we want to create;
  - m lines with a 3-letter string y_j, the type of the jth amino acid molecule in the protein chain we want to create.

  In a given case, no more than 20 different amino acid molecules will appear.

H.2 Output

For each test case, the output contains one line with one integer number: the cost (in euros) of converting a chain of the protein $(x_1, x_2, \ldots, x_k)$ into a chain of the protein $(y_1, y_2, \ldots, y_m)$.

H.3 Example

Input	Output
2	5
2	12
Lys	
Trp	
3	
Lys	
His	
Trp	
6	
Ala	
Pro	
Pro	
Val	
Phe	
Met	
5	
Ala	
Ser	
Ser	
Val	
Met	

Table 8: Example input and output