

SAPIENTIA HUNGARIAN UNIVERSITY OF TRANSYLVANIA



ECN INTERNATIONAL PROGRAMMING CONTEST 2016

Problems



Târgu Mureş / Marosvásárhely, 21th May 2016

Problem A

Input file :	A.IN
Output file :	standard output

The students are preparing for the national examination. Until last year, this exam consisted of two mandatory tests. The new minister of Education introduces the third mandatory test this year. The computer scientist of the Ministry does not intend to alter the original database structure, so he creates a new table for the third test.

The input file contains the following data. The first line contains the number of students, $n < 250000$. The following n lines contain the first part of the data about the exam (conforming to the old law): personal code of student (31 bits positive integer), name of the first test, evaluation of the first test, name of the second test and evaluation of the second test. The last n lines contain the data about the third test, recently added: personal code of student, test name and test evaluation.

The name of the students and any other related data are stored in another table; we are not interested about its structure.

Every test name is a string of at most 35 characters without space. A test is evaluated with a number of points between 1 (1 point is just for presence) and 100, or A if the student is absent, or E if the student is eliminated due to a tentative of fraud.

A test is considered to be passed if its evaluation is at least 50 points. The student passes the national exam if the average of the three tests is at least 60 points, all these tests must be passed. If any test is marked with E, the student is eliminated. If any test is marked with A and no test is marked with E, the student is absent. Show the statistics of the exam using the following pattern.

Sample input: A.IN	Sample output: standard output
6	
10 Inf 70 Mat 60	Total students:
2 Law 60 Eco A	6
5 Inf 40 Mat 80	Passed: 2 33.33%
12 Law 55 Eco 64	Not passed: 4 66.67%
15 Hst 90 Geo E	Absents: 1 16.67%
8 Psy 56 Geo 54	Eliminated: 1 16.67%
12 Hst 75	
10 Psy 80	
8 Hst 64	
5 Eco 60	
2 Hst 50	
15 Psy A	

Problem B

$$(51! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot 51 = \dots 0000000000000, 51/5 = 10/5 = 2/5 = 0)$$

Input file :	B.IN
Output file :	standard output

Write a program to find the number of trailing zeros at the end of $k!$ ($k < 2^5 \cdot 10^3$ factorial).

Input

The first line of the input contains n , the number of test cases. This is followed by n lines, each containing a number.

Output

For each number your program has to print on a separate line the number of trailing zeros.

Sample input: B.IN	Sample output: standard output
5	1
7	2
10	2
13	24
100	51
212	

Problem C

Input file :	C.IN
Output file :	standard output

From the input file a number "count" is read, then 2 arrays of strings with "count" members each. The first array of strings sets the initial order of the strings, while the second array is a permutation of the first array (same strings, but in a different order).

Requirement:

Find the longest subsequence of strings in the second array which has the property that has the words in the same order as in the first array. The output file must contain the maximum number of strings and then the strings. If there are more subarrays with maximum length, it will display one of them.

Sample input: C.IN	Sample output: standard output
6 madrid munchen oslo rome budapest bucharest oslo rome munchen madrid bucharest budapest	3 oslo rome Bucharest

Problem D

Input file :	D.IN
Output file :	standard output

Imagine a simplified biological world (in 2D), where the proteins in the cells are still linear chains of amino-acids, folded in as compact shape as it is possible (see the detailed description below). It is known that the functionality of proteins is mostly determined by their geometric shape, at the end of the folding process.

Let us suppose the amino-acids are linear strait segments of length 1,2,3,...,10. The first five are sketched in the figure on the right. They can only stay horizontally (as in the figure) or vertically.

Each one has a tail, marked with black, and a head on the opposite end, which can establish a chemical bond with the tail of on another amino-acid in one of the directions ahead, left or right, as shown in this figure. Notice, that there are no possible shapes of other forms, such as in the next figure.

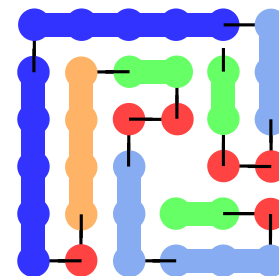
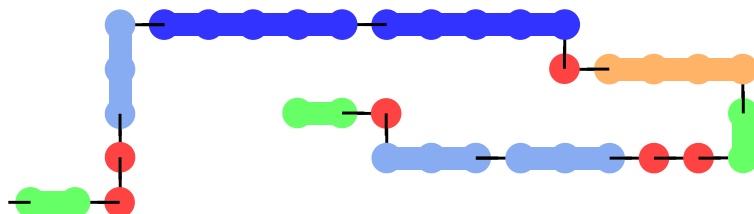
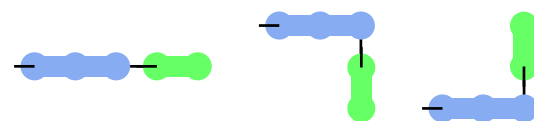
A chain of amino-acids can be consequently simply described by a sequence of numbers. As an example the sequence 2, 1, 1, 3, 5, 5, 1, 4, 2, 1, 1, 3, 3, 1, 2 represents the chain shown in the figure below.

It is obvious that the spatial shape depicted here is quite arbitrary, it is not yet determined, as it is not the most compact spatial arrangement. That's just the challenge: find the most compact folding! Suppose the measure of the compactness is just the size of the bounding rectangle, more precisely its perimeter (actually the half perimeter can be considered).

The smaller the half perimeter is, the folding is better. In the example above (one of) the best folding is shown in the next figure. The measure of compactness, i.e. the size of the bounding rectangle is $6+6=12$, as it can be easily seen on the figure. This is the best possible achievement. This can be indirectly deduced from several observations. First the total length of the chain is 35, which in the extrem case can be packed in a rectangle of 5×7 , giving the same measure $5+7=12$. It is known that the minimum perimeter of a rectangle of a given area (35 in our case) is attained for the shape of the square, therefore we look for a rectangle as close as possible to a square.

The adjacent pieces of length 5 must therefore be considered in an L shape. The minimum side-length of a box containing a piece of length 5 is 6, so a bounding square of side length 6 is a minimal one.

Your task is to compose an algorithm and a program to fold a protein given by a sequence of integer numbers, each between 1 and 10. The maximum number of terms in the sequence is 1000. You have to search for a folding as compact as you can. The measure of compactness is the half perimeter of the bounding rectangle.



Input file

The input file contains in the first row an integer n , the number of test cases, followed by n rows, each containing a sequence of comma separated numbers, of a length which not exceeds 1000 terms. The numbers in these rows are between 1 and 10.

Output file

This file should contain n pairs of rows. The first row of the pair contains three numbers separated by comma, the first one – the size of the bounding box – being the sum of the other two (the rectangle side-lengthes), representing the dimensions of the minimum size bounding box found by the algorithm for the corresponding protein from the input list. The second row of the pair must contain the description of the folding: this would be a sequence of comma separated letters, U,D,L,R, (up, down, left, right) meaning the position of the corresponding amino-acid in the space (see the example). In the case for a protein the algorithm did not finishes the search in 7 seconds, two lines containing each the character "-" should be inserted.

Example: Using the protein given in the text of the problem, the input and the corresponding output file is:

Sample input: D.IN	Sample Output: standard output
1	12,6,6
2,1,1,3,5,5,1,4,2,1,1,3,3,1,2	D,D,R,U,L,D,R,U,R,D,L,D,R,U,L

Problem E

Input file :	E.IN
Output file :	standard output

Adele is a French typist. Since French people tend to talk very fast, on her first days she struggled a lot writing down correctly all what was dictated. So in order to be faster, she decided to write for each sentence the words together, without the blanks (spaces). After the dictation ended she went home and wanted to split the sentences back into words. Since she has to use a corrector, she wants to separate the sentences into the fewer number of words possible. This is a laborious task; your job is to help Adele.

Input:

- First line will contain t , the number of test cases
- For each test case
 - the first line will contain the number of words in the dictionary n followed by a blank and the concatenated sentence s .
 - the next n lines will contain the words from the vocabulary

Input limits

- $0 < t \leq 10$, $0 < n \leq 32700$, $0 < \text{length of } s \leq 32700$,
- words and sentences contain only small letters of the Latin alphabet
- Note: the longest French word has 25 letters: anticonstitutionnellement

Output:

- Should contain t lines
- For each test case:
 - The output line should contain the minimal number of words in which the sentence can be split, followed by the split sentence (if it is possible to split optimally the sentence in multiple ways than any of the solutions is accepted)
 - If there is no solution a 0 should be printed

Example:

Sample input: E.IN	Sample Output: standard output
3, 6 toutlemondeestbeau, tout, le, lemon, monde, est, beau, 7 onelogicfr, fr, one, onelogi, c, f, r, logic, 3 pasdesolution, pas, des, solution	5 tout le monde est beau, 3 one logic fr, 0

Problem F

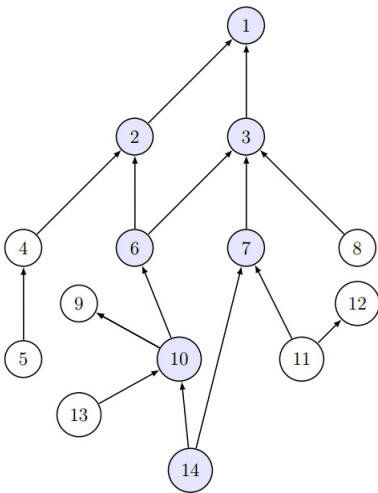
Input file :	F.IN
Output file :	standard output

Let G be a connected directed graph with no directed cycle. A subgraph H of G is *fishgraph*, if the following are true:

- H is connected,
- H has only one *source vertex*, that has no ingoing arcs,
- H has only one *sink vertex*, that has no outgoing arcs,
- each vertex of H different from source and sink has ingoing and outgoing arcs too.

Each directed path in G is a trivial fishgraph. Nontrivial fishgraphs are those for which the source has at least two outgoing arcs and the sink at least two ingoing arcs.

Example:



In this graph the only nontrivial fishgraphs are:

- the subgraph spanned by vertices 1, 2, 3, 6, 7, 10, and 14,
- the subgraph spanned by vertices 1, 2, 3, and 6,
- the subgraph spanned by vertices 3, 6, 7, 10, and 14.

The task is to find a fishgraph with source x and sink y , if such a fishgraph exists.

Input

- The first line contains the number of vertices in G .
- The second line contains the number m of arcs in G .
- Each of the next m lines contains the two vertices of an arc separated by a space.
- The next line contains the number r of tests.
- Each of the next r lines contains a source and a sink vertex corresponding to a test.

Output

For the i th test-case

- print in a line the text: **Test i** ,
- print in the next line the vertices in increasing order of the corresponding fishgraph separated by a space. If such a fishgraph does not exist, print the word **NO**.

Restrictions and refinements

- For the number n of vertices: $1 \leq n \leq 100$.
- The vertices of the graph are natural numbers from 1 to n .

Sample input: F.IN	Sample Output: standard output
14	
15	
14 10	
14 7	
13 10	
11 12	
11 7	
10 9	Test 1
10 6	1 2 3 6 7 10 14
8 3	Test 2
7 3	1 2 3 6
6 2	Test 3
6 3	NO
5 4	Test 4
4 2	3 6 7 10 14
2 1	
3 1	
4	
14 1	
6 1	
12 7	
14 3	

Problem G

Input file :	G.IN
Output file :	standard output

Let P be a convex polygon with N vertices: $(x[i], y[i])$ where both coordinates of each point are natural numbers. You can rotate the polygon with any degree once. After the rotation take the lowest point on the polygon: the one with the smallest $y[i]$, let that value be y_{\min} . All the points that have $y[i]$ values (after the rotation) in the range: $[y_{\min}, y_{\min} + L]$ are considered lucky, where L is a given natural number. What is the maximum number of lucky vertices you can make by properly rotating the polygon?

Input

The first line contains T , the number of test cases. Each test case description follows in order: The first line contains N and L , the number of vertices and the width L . The next N lines contain two natural numbers $x[i]$ and $y[i]$, the initial coordinates of the points in order. The order can be counterclockwise or clockwise.

Output

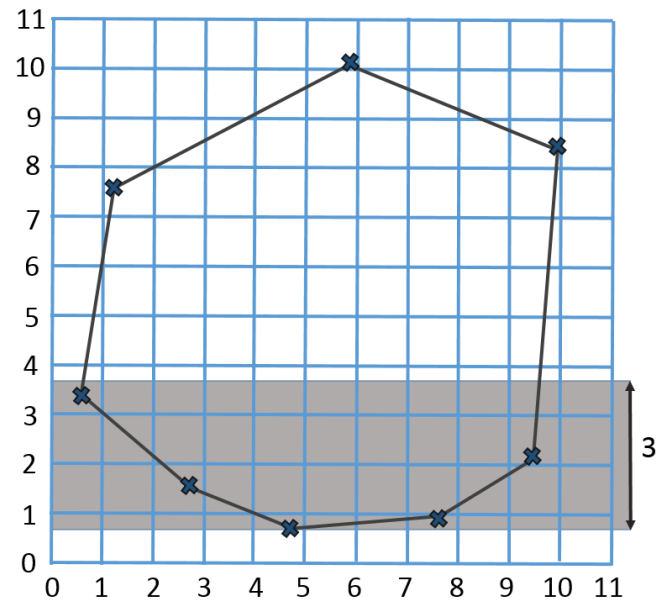
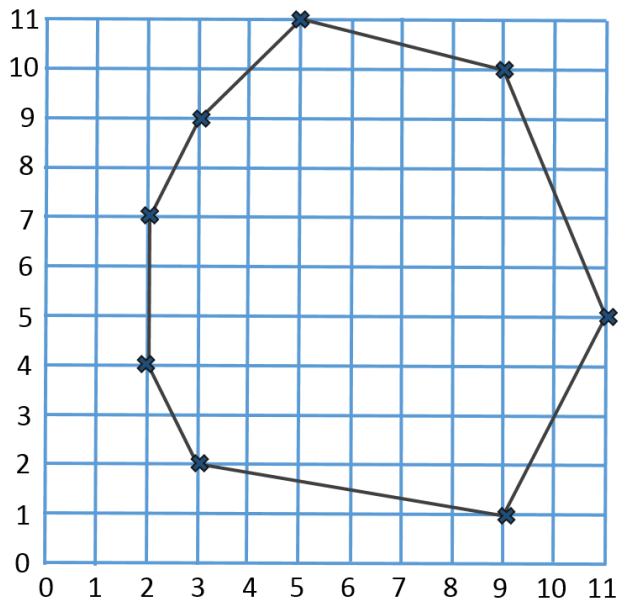
For each test case print the maximum number of points that can be lucky after the rotation.

Restrictions and refinements

- $1 \leq t \leq 10$,
- $1 \leq N \leq 10^5$,
- $0 \leq x[i], y[i], L \leq 10^9$
- The rotation can be made by a non-integer amount of degrees: you can rotate with any precision.
- The rotation center is not relevant because lucky points are relative to the lowest point.

Sample input: G.IN	Sample Output: standard output
1	5
8 3	
9 1	
11 5	
9 10	
5 11	
3 9	
2 7	
2 4	
3 2	

Details: Points with initial coordinates (5, 11), (3, 9), (2, 7), (2, 4), (3, 2) can be lucky.



Problem H

Input file :	H.IN
Output file :	standard output

Each card in a deck has an uppercase letter of the English alphabet on one side, and an integer number on the other side. There is exactly one letter and one number on each card. Somebody makes the following statement about this card deck:

"If there is a vowel on the letter side of a card, then its number side shows an even number."

To determine whether this statement is true, we obviously have to have a look at the other side of certain cards.

Your task is to write a program that

1. reads data describing one side of the cards,
2. computes the minimum number of cards that have to be flipped over in the worst case to be able to determine without any doubt whether the above statement is true for all cards in the deck,
3. and prints the result.

Input

The input consists of multiple test cases. Each line of a test case contains either an uppercase letter of the English alphabet or an integer. Test cases are terminated by a line containing the string "END". You may assume that each test case contains at least one line describing a card.

Output

For each test case, your program should output one line containing one integer: the minimum number of cards that need to be flipped over.

Sample input: H.IN	Sample Output: standard output
E	
K	
4	2
7	
END	

Problem I

Input file :	I.IN
Output file :	standard output

Let's imagine you are the system administrator of a critical IT infrastructure composed of multiple servers running different software. Suddenly you notice that multiple components of your system are affected by different software bugs, n ($n \leq 20$) bugs in total. In this situation the whole system becomes vulnerable to attacks, and this situation persists while you fix all the bugs.

You receive the exact instructions from the support team about the way to fix the bugs, this way you are able to estimate the required solving time ($\sigma \leq 10^3$) separately for all the bugs. In addition, all the bugs are ranked by the security team, therefore each of them has an initial risk score assigned (r): 1-low, 2-medium, 3-high.

Unfortunately, time is your enemy: the risk scores are increasing direct-proportionally with the time spent as long as they are exploitable. The time dependent risk (η) for the i^{th} bug can be calculated as follows:

$$\eta_i(t) = \begin{cases} 0, & t < 0 \text{ or } t > \tau_i, \\ r_i \cdot t, & \text{otherwise} \end{cases}$$

where τ_i represents the end time of the treating for the i^{th} bug.

Your job is to schedule the solving of the bugs in the optimal order, more precisely to find that order which minimize the integral of the cumulative risk (q) for the whole intervention period, where

$$q(t) = \sum_{i=1}^n \eta_i(t),$$

$$\int_0^t q(x)dx = \sum_{i=1}^n r_i \cdot \frac{\left(\sum_{k=1}^i \sigma_k\right)^2}{2}.$$

Input

The first line of the input file contains the number of the given test cases, after which each line contains a test case: pairs of integers representing the initial risk scores and the solving times required for the bugs ($r_1, \sigma_1, r_2, \sigma_2, ..$) separated with space characters.

Output

A line for each test case written to the standard output: the indexes of the risk-time pairs in the correct order separated with space characters.

Example

Sample input: I.IN	Sample Output: standard output
1 3 10 2 11 1 5 3 11 2 5 1 10	5 1 4 3 2 6

Details:

$$\begin{aligned}
 \int_0^t q(x)dx &= 2 \cdot \frac{5^2}{2} + 3 \cdot \frac{(5+10)^2}{2} + 3 \cdot \frac{(5+10+11)^2}{2} + 1 \cdot \frac{(5+10+11+5)^2}{2} \\
 &\quad + 2 \cdot \frac{(5+10+11+5+11)^2}{2} + 1 \cdot \frac{(5+10+11+5+11+10)^2}{2} \\
 &= 4973
 \end{aligned}$$

Problem J

Input file :	J.IN
Output file :	standard output

The settlements of a county are connected with a network of two-way roads. In order to reach every settlement from the county town with regular bus services, county roads must be properly consolidated. But, as county budget is severely cut, they are forced to spend as less as possible for the construction.

Write a program that calculates which are those sections that certainly won't be consolidated!

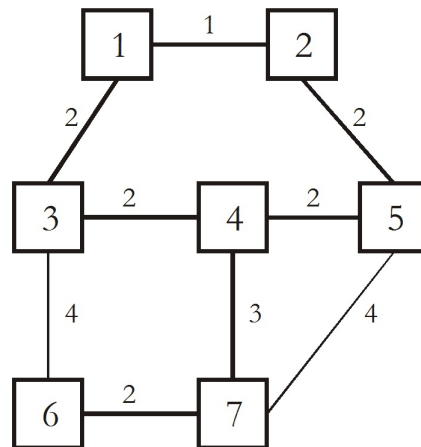
Input

The first line of the input file contains the number of the given test cases, the next line of the input file is given the number of the settlements to be connected ($2 \leq N \leq 10000$) and the number of the existing road sections ($1 \leq M \leq 500000$). The next M lines contain three integer numbers: the road descriptions, each line consisting of the number of two settlements ($1 \leq A_i \neq B_i \leq N$) and the consolidation costs involved for the section ($1 \leq K_i \leq 1000$).

Output

The first line of the standard output will contain the total number U of the sections which will not be consolidated. The second line contains the increasingly ordered number of the sections that won't be consolidated, as they are enumerated in the input file. Each road number will separated with one space from its neighbours.

Example:



Sample input: J.IN	Sample Output: standard output
7 9	
1 2 1	
1 3 2	
2 5 2	
3 4 2	2
5 4 2	6 9
5 7 4	
7 4 3	
6 7 2	
3 6 4	

Problem K

Input file :	K.IN
Output file :	standard output

Imagine a k -dimensional array, $c[1..n][1..n] \dots [1..n]$ ($k > 0, n > 0, n^k < 10^6$). The integer values ($-10^{10}..10^{10}$) the cells store are given in the following way (all of them are well defined):

- To each cell a string is assigned.
- These strings may contain positive decimal integers, the MIN/MAX/ADD/MUL functions or cell-references. (MIN/MAX/ADD/MUL represents the minima, maxima, sum, and product functions respectively)
- Each function has the following syntax:
`funcionID(argument_number:argument-1,argument-2,...,argument-argument_number)`
 where `argument_number` is a positive decimal integer and the arguments may contain positive decimal integers, the MIN/MAX/ADD/MUL functions or cell-references.
- The cell-references have the following syntax: `c[index_value-1][index_value-2]...[index_value-k]`
 where the `index_values` are integers between 1 and n .

Your task is to find the value that a given target cell stores.

Input

The first line of the input file contains the number t of test cases. The first line of each test case includes the k and n (separated with a space). Each of next n^k line includes the cell-indexes of the current cell, and the string we assigned to corresponding cell (separated by spaces). The last line contains the indexes of the target-cell (separated by spaces).

Output

For each test case your program has to print on separate lines the value the target cell stores.

Sample input: K.IN	Sample Output: standard output
<pre> 1 2 3 1 1 MAX(2:ADD(2:7,c[2][1]),ADD(2:7,c[2][2])) 1 2 0 1 3 0 2 1 MAX(2:ADD(2:5,c[3][1]),ADD(2:5,c[3][2])) 2 2 MAX(2:ADD(2:9,c[3][2]),ADD(2:9,c[3][3])) 2 3 0 3 1 15 3 2 2 3 3 5 1 1 </pre>	<p>27</p>

Problem L

Input file :	L.IN
Output file :	standard output

John Doe's house is a big mess now, due to yesterday's party. His watch is in the refrigerator, there are some oranges in the bed and cigarettes in the microwave oven. He wants to clean up as fast as possible, because his mother is coming to visit him soon, and the landscape isn't exactly what John wants his mother to see.

John's house has n rooms and can be described as an undirected graph with n nodes and m edges. John is in the bedroom (numbered 1) and wants to end the cleaning process in the hall (numbered n). There are k objects which need to be placed back to their original place. For each object we know its current place, and the number of the room in which it must end up. John can carry at most p objects at one time. He can make the following operations:

- Move to a neighboring room.
- Pick up an object that is in the same room as him.
- Leave an object he is carrying in the current room, if it is the destination room of the object.

Any of these operations takes exactly thirteen seconds.

Input

The first line contains l , the number of cases. Each case is described as follows. The first line contains n and m . m lines follow, each containing the indices of two neighboring rooms. The following line contains k and p . The following k lines contain the current location and the final location of an object. Empty lines can appear freely.

Output

The output should contain the number of people that have partied yesterday, John's age, and his mother's maiden name. Ok, forget about that, and just write out the number of seconds needed to clean up :) Each answer should be on a separate line.

Restrictions

- $1 \leq l \leq 15$
- $1 \leq n \leq 50$
- $0 \leq m \leq 1000$
- $0 \leq k \leq 5$
- $1 \leq p \leq 5$
- The graph will be conex.

Example

Sample input: L.IN	Sample Output: standard output
1	
4 3	
1 2	
2 3	65
4 3	
1 2	
2 3	

John makes the following operations: moves from room 1 to room 2, picks up the object in room 2, moves from room 2 to room 3, leaves the object in room 3, moves from room 3 to room 4.