

The 17th Zhejiang University Programming Contest

Sponsored by



Contest Session

April 22, 2017



Problem List

A	Marjar Cola
B	Edge to the Root
C	How Many Nines
D	Independent Set
E	Fibonacci Sequence Chicken Edition
F	Intervals
G	Seven-Segment Display
H	Saddle Point
I	Course Selection System
J	Knuth-Morris-Pratt Algorithm

This problem set should contain 10 (ten) problems on 13 (thirteen) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.

<https://sua.ac/>

Hosted by



Problem Set Prepared by

签到成功 这是你的
签到奖励



It's against the rules to open non-contest websites during the contest.
If you're interested (which is our pleasure), please scan the
QR code only after the contest.

Problem A. Marjar Cola

Marjar Cola is on sale now! In order to attract more customers, Edward, the boss of Marjar Company, decides to launch a promotion: If a customer returns x empty cola bottles or y cola bottle caps to the company, he can get a full bottle of Marjar Cola for free!

Now, Alice has a empty cola bottles and b cola bottle caps, and she wants to drink as many bottles of cola as possible. Do you know how many full bottles of Marjar Cola she can drink?

Note that a bottle of cola consists of one cola bottle and one bottle cap.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \leq T \leq 100$), indicating the number of test cases. For each test case:

The first and only line contains four integers x, y, a, b ($1 \leq x, y, a, b \leq 100$). Their meanings are described above.

Output

For each test case, print one line containing one integer, indicating the number of bottles of cola Alice can drink. If Alice can drink an infinite number of bottles of cola, print “INF” (without quotes) instead.

Example

standard input	standard output
2	INF
1 3 1 1	4
4 3 6 4	

Note

For the second sample test case, Alice has 6 empty bottles and 4 bottle caps in hand. She can return 4 bottles and 3 caps to the company to get 2 full bottles of cola. Then she will have 4 empty bottles and 3 caps in hand. She can return them to the company again and get another 2 full bottles of cola. This time she has 2 bottles and 2 caps in hand, but they are not enough to make the exchange. So the answer is 4.

Problem B. Edge to the Root

Given a tree with n vertices, we want to add an edge between vertex 1 and vertex x , so that the sum of $d(1, v)$ for all vertices v in the tree is minimized, where $d(u, v)$ is the minimum number of edges needed to pass from vertex u to vertex v . Do you know which vertex x we should choose?

Recall that a tree is an undirected connected graph with n vertices and $(n - 1)$ edges.

Input

There are multiple test cases. The first line of input contains an integer T , indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 2 \times 10^5$), indicating the number of vertices in the tree.

Each of the following $n - 1$ lines contains two integers u and v ($1 \leq u, v \leq n$), indicating that there is an edge between vertex u and v in the tree.

It is guaranteed that the given graph is a tree, and the sum of n over all test cases does not exceed 5×10^5 . As the stack space of the online judge system is not very large, the maximum depth of the input tree is limited to about 3×10^4 .

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use `scanf/printf` instead of `cin/cout` in C++.

Output

For each test case, output a single integer indicating the minimum sum of $d(1, v)$ for all vertices v in the tree (NOT the vertex x you choose).

Example

standard input	standard output
2	8
6	2
1 2	
2 3	
3 4	
3 5	
3 6	
3	
1 2	
2 3	

Note

For the first test case, if we choose $x = 3$, we will have

$$d(1, 1) + d(1, 2) + d(1, 3) + d(1, 4) + d(1, 5) + d(1, 6) = 0 + 1 + 1 + 2 + 2 + 2 = 8$$

It's easy to prove that this is the smallest sum we can achieve.

Problem C. How Many Nines

If we represent a date in the format YYYY-MM-DD (for example, 2017-04-09), do you know how many 9s will appear in all the dates between $Y_1-M_1-D_1$ and $Y_2-M_2-D_2$ (both inclusive)?

Note that you should take leap years into consideration. A leap year is a year which can be divided by 400 or can be divided by 4 but can't be divided by 100.

Input

The first line of the input is an integer T ($1 \leq T \leq 10^5$), indicating the number of test cases. Then T test cases follow. For each test case:

The first and only line contains six integers Y_1 , M_1 , D_1 , Y_2 , M_2 , and D_2 , their meanings are described above.

It's guaranteed that $Y_1-M_1-D_1$ is not larger than $Y_2-M_2-D_2$. Both $Y_1-M_1-D_1$ and $Y_2-M_2-D_2$ are between 2000-01-01 and 9999-12-31, and both dates are valid.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use `scanf/printf` instead of `cin/cout` in C++.

Output

For each test case, you should output one line containing one integer, indicating the answer of this test case.

Example

standard input	standard output
4	4
2017 04 09 2017 05 09	2
2100 02 01 2100 03 01	93
9996 02 01 9996 03 01	1763534
2000 01 01 9999 12 31	

Note

For the first test case, four 9s appear in all the dates between 2017-04-09 and 2017-05-09. They are: 2017-04-09 (one 9), 2017-04-19 (one 9), 2017-04-29 (one 9), and 2017-05-09 (one 9).

For the second test case, as year 2100 is not a leap year, only two 9s appear in all the dates between 2100-02-01 and 2100-03-01. They are: 2017-02-09 (one 9) and 2017-02-19 (one 9).

For the third test case, at least three 9s appear in each date between 9996-02-01 and 9996-03-01. Also, there are three additional nines, namely 9996-02-09 (one 9), 9996-02-19 (one 9) and 9996-02-29 (one 9). So the answer is $3 \times 30 + 3 = 93$.

Problem D. Independent Set

Chiaki has a positive integer m and she would like to construct a tree with at most 15 vertices in such a manner that the number of distinct non-empty independent sets is exactly m .

Note that an independent set is a subset of vertices of a graph such that every two distinct vertices are not adjacent.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \leq T \leq 2000$), indicating the number of test cases. For each test case:

The first line contains an integer m ($1 \leq m \leq 2000$).

Output

For each test case, if there is no such graph, output -1 on a single line. Otherwise, output an integer n ($1 \leq n \leq 15$) denoting the number of vertices. Then in each of the next $(n - 1)$ lines, output two integers x and y ($1 \leq x, y \leq n$, $x \neq y$) denoting an edge in the tree.

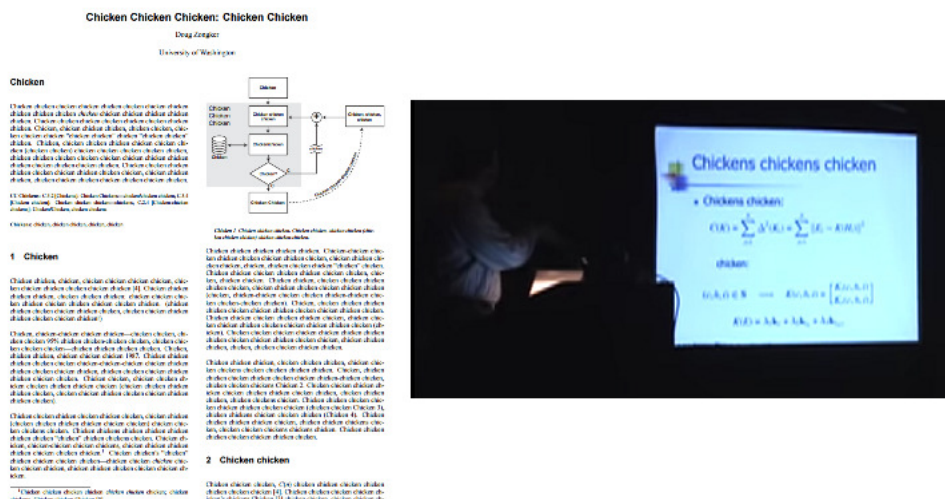
Example

standard input	standard output
5	1
1	2
2	2 1
3	-1
10	-1
20	6
	2 1
	3 1
	4 3
	5 4
	6 5

Problem E. Fibonacci Sequence Chicken Edition

Year 2017 is the year of chicken, so in this problem we introduce you an interesting programming language: the Chicken Language.

Chicken is an esoteric programming language by Torbjörn Söderstedt, in which “chicken” is the only valid symbol. It is inspired by the paper and the presentation presented at the AAAS humor session by Doug Zongke. We strongly recommend you to watch the presentation after the contest. It’s super fun.



The paper and the presentation by Doug Zongke.

From https://www.youtube.com/watch?v=yL_-1d9OSdk

As the original chicken language is a bit complicated, we specially designed the *Simplified Chicken Language* (SCL) for this problem. An SCL program is consisted of only two kinds of tokens: “c” and new line. The number of “c” tokens in the same line corresponds to an opcode. As the program is executed, it will push/pop values to/from a stack (the stack is empty at the beginning). The opcodes and the descriptions for each instruction of SCL are listed below.

Please keep in mind that the stack of the SCL is 1-based. That is to say, the index of the bottom of the stack is considered to be 1. In the following table, we indicate the integer at the top of the stack as x , and the integer just below the top of the stack as y . We also indicate $stack[n]$ as the integer in the stack whose index is n .

Opcode	Instruction	Description
1	Add	Pop x and y , and push $x + y$ back to the stack. If there are less than two integers in the stack, the program will crash.
2	Subtract	Pop x and y , and push $x - y$ back to the stack. If there are less than two integers in the stack, the program will crash.
3	Compare	Pop x and y . If x equals to y then push 1 onto the stack, otherwise push 0 onto the stack. If there are less than two integers in the stack, the program will crash.
4	Load	Load one integer from the standard input and push it onto the stack. If there is nothing to load, the program will crash.
5	Copy	Pop x and y , and copy $stack[y]$ to $stack[x]$. If there are less than two integers in the stack, the program will crash. If x or y is out of the range of the current stack, the program will also crash.
6	Jump	Pop x and y . If y is not zero, the program will jump to the x -th line and continue to run. If x is larger than the number of lines of the program, the program will stop. If there are less than two integers in the stack, the program will crash. If x is not a positive integer, the program will also crash.
7+	Push	If the Opcode is larger or equal to 7, push the integer ($Opcode - 7$) onto the stack.

When the program stops, it will print out the integer at the top of the stack as the output.

What you need to do is to write an SCL program which can print out the n -th element of the Fibonacci sequence. Recall that a Fibonacci sequence is a sequence which satisfies $f(1) = f(2) = 1$ and $f(n) = f(n-1) + f(n-2)$ when $n \geq 3$.

As the online judge system of the contest does not support SCL, you're supposed to print out your SCL program using other languages like C or C++. A specially designed program will then judge the correctness of your output.

Input

There is no input for your C/C++/etc. program.

For your SCL program, only one integer n ($1 \leq n \leq 30$) will be given in the standard input, indicating the index of the element in the Fibonacci sequence.

Output

Your C/C++/etc. program should output your SCL program.

Please output your SCL program in the correct format. If your program contains tokens other than "c" or new line (for example, space), or you print extra empty lines at the end of your output, you will get a "wrong answer" verdict.

Also, your SCL program should contain no more than 10^4 "c" tokens, and your SCL program can execute (NOT contain, but execute!) at most 10^3 lines of code for each SCL test case, or you will get a "wrong answer" verdict.

Note that if your SCL program crashes due to various reasons, you will still get a "wrong answer" verdict as this problem is special judged.

Your SCL program should print out only one integer $f(n)$, which is the n -th element of the Fibonacci sequence.

Example

standard input	standard output
(No input for your C/C++/etc. program)	cccc ccccccc cccccccc cccccccc cccc cccccccc ccc cccccccccccccccccccc cccccc cccccccc c

Note

Please note that the sample output is only to show you the output format, and it is NOT A CORRECT ANSWER!

What the sample SCL program do is to read an integer from the standard input and check if the integer equals to 2. If it does, then print out the integer directly, otherwise add 2 to the integer and print it out. You can try to figure out how this SCL program works, as it may help you to understand the problem better.

Problem F. Intervals

Chiaki has n intervals and the i -th of them is $[l_i, r_i]$. She wants to delete some intervals so that there does not exist three intervals a , b and c such that a intersects with b , b intersects with c and c intersects with a .

Chiaki is interested in the minimum number of intervals which need to be deleted.

Note that interval a intersects with interval b if there exists a real number x such that $l_a \leq x \leq r_a$ and $l_b \leq x \leq r_b$.

Input

There are multiple test cases. The first line of input contains an integer T , indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 5 \times 10^4$) – the number of intervals.

Each of the following n lines contains two integers l_i and r_i ($1 \leq l_i < r_i \leq 10^9$) denoting the i -th interval. Note that for every $1 \leq i < j \leq n$, $l_i \neq l_j$ or $r_i \neq r_j$.

It is guaranteed that the sum of all n does not exceed 5×10^5 .

Output

For each test case, first output an integer m denoting the minimum number of deletions. Then output m integers in increasing order denoting the index of the intervals to be deleted.

Example

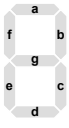
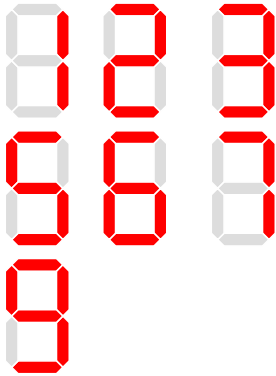
standard input	standard output
1 11 2 5 4 7 3 9 6 11 1 12 10 15 8 17 13 18 16 20 14 21 19 22	4 3 5 7 10

Problem G. Seven-Segment Display

A seven segment display, or seven segment indicator, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays. Seven segment displays are widely used in digital clocks, electronic meters, basic calculators, and other electronic devices that display numerical information.

The segments of a seven segment display are arranged as a rectangle of two vertical segments on each side with one horizontal segment on the top, middle, and bottom. If we refer the segments as the letters from a to g , it's possible to use the status of the segments which is called a seven segment code to represent a number. A standard combination of the seven segment codes is shown below.

X	a	b	c	d	e	f	g
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0

0 = on
1 = off

A seven segment code of permutation p is a set of seven segment code derived from the standard code by rearranging the bits into the order indicated by p . For example, the seven segment codes of permutation “gbedcfa” which is derived from the standard code by exchanging the bits representing by “a” and “g”, and by exchanging the bits representing by “c” and “e”, is listed as follows.

X	g	b	e	d	c	f	a
1	1	0	1	1	0	1	1
2	0	0	0	0	1	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	0	0	1
5	0	1	1	0	0	0	0
6	0	1	0	0	0	0	0
7	1	0	1	1	0	1	0
8	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0

We indicate the seven segment code of permutation p representing number x as $c_{p,x}$. For example $c_{abcdefg,7} = 0001111$, and $c_{gbedcfa,7} = 1011010$.

Given n seven segment codes s_1, s_2, \dots, s_n and the numbers x_1, x_2, \dots, x_n each of them represents, can you find a permutation p , so that for all $1 \leq i \leq n$, $s_i = c_{p,x_i}$ holds?

Input

The first line of the input is an integer T ($1 \leq T \leq 10^5$), indicating the number of test cases. Then T test cases follow.

The first line of each test case contains an integer n ($1 \leq n \leq 9$), indicating the number of seven segment codes.

For the next n lines, the i -th line contains a number x_i ($1 \leq x_i \leq 9$) and a seven segment code s_i ($|s_i| = 7$), their meanings are described above.

It is guaranteed that $\forall 1 \leq i < j \leq n, x_i \neq x_j$ holds for each test case.

Output

For each test case, output “YES” (without the quotes) if the permutation p exists. Otherwise output “NO” (without the quotes).

Example

standard input	standard output
3	YES
9	NO
1 1001111	YES
2 0010010	
3 0000110	
4 1001100	
5 0100100	
6 0100000	
7 0001111	
8 0000000	
9 0000100	
2	
1 1001111	
7 1010011	
2	
7 0101011	
1 1101011	

Note

For the first test case, it is a standard combination of the seven segment codes.

For the second test case, we can easily discover that the permutation p does not exist, as three in seven bits are different between the seven segment codes of 1 and 7.

For the third test case, $p = \text{agbfcd}$.

Problem H. Saddle Point

Chiaki has an $n \times m$ matrix A . Rows are numbered from 1 to n from top to bottom and columns are numbered from 1 to m from left to right. The element in the i -th row and the j -th column is $A_{i,j}$.

Let $M(\{i_1, i_2, \dots, i_s\}, \{j_1, j_2, \dots, j_t\})$ be the matrix that results from deleting row i_1, i_2, \dots, i_s and column j_1, j_2, \dots, j_t of A and $f(\{i_1, i_2, \dots, i_s\}, \{j_1, j_2, \dots, j_t\})$ be the number of saddle points in matrix $M(\{i_1, i_2, \dots, i_s\}, \{j_1, j_2, \dots, j_t\})$.

Chiaki would like to find all the value of $f(\{i_1, i_2, \dots, i_s\}, \{j_1, j_2, \dots, j_t\})$. As the output may be very large $((2^n - 1)(2^m - 1))$ matrix in total, she is only interested in the value

$$\sum_{\substack{1 \leq i_1 < \dots < i_s \leq n \\ 1 \leq j_1 < \dots < j_t \leq m \\ 0 \leq s < n \\ 0 \leq t < m}} f(\{i_1, i_2, \dots, i_s\}, \{j_1, j_2, \dots, j_t\}) \mod (10^9 + 7)$$

Note that a saddle point of a matrix is an element which is both the only largest element in its column and the only smallest element in its row.

Input

There are multiple test cases. The first line of input contains an integer T , indicating the number of test cases. For each test case:

The first line contains four integers n and m ($1 \leq n, m \leq 1000$) – the number of rows and the number of columns.

Each of the next n lines contains m integer $A_{i,1}, A_{i,2}, \dots, A_{i,m}$ ($1 \leq A_{i,j} \leq 10^6$), where $A_{i,j}$ is the integer in the i -th row and the j -th column.

It is guaranteed that neither the sum of all n nor the sum of all m exceeds 5000.

Output

For each test case, output an integer denoting the answer.

Example

standard input	standard output
2	4
2 2	465
1 1	
1 1	
4 5	
1 2 3 4 5	
6 7 8 9 10	
11 12 13 14 15	
16 17 18 19 20	

Problem I. Course Selection System

There are n courses in the course selection system of Marjar University. The i -th course is described by two values: happiness H_i and credit C_i . If a student selects m courses x_1, x_2, \dots, x_m , then his comfort level of the semester can be defined as follows:

$$\left(\sum_{i=1}^m H_{x_i}\right)^2 - \left(\sum_{i=1}^m H_{x_i}\right) \times \left(\sum_{i=1}^m C_{x_i}\right) - \left(\sum_{i=1}^m C_{x_i}\right)^2$$

Edward, a student in Marjar University, wants to select some courses (also he can select no courses, then his comfort level is 0) to maximize his comfort level. Can you help him?

Input

There are multiple test cases. The first line of input contains an integer T , indicating the number of test cases. For each test case:

The first line contains a integer n ($1 \leq n \leq 500$) – the number of courses.

Each of the next n lines contains two integers H_i and C_i ($1 \leq H_i \leq 10^4, 1 \leq C_i \leq 100$).

It is guaranteed that the sum of all n does not exceed 5000.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use `scanf/printf` instead of `cin/cout` in C++.

Output

For each case, you should output one integer denoting the maximum comfort.

Example

standard input	standard output
2	191
3	0
10 1	
5 1	
2 10	
2	
1 10	
2 10	

Note

For the first case, Edward should select the first and second courses.

For the second case, Edward should select no courses.

Problem J. Knuth-Morris-Pratt Algorithm

In computer science, the Knuth-Morris-Pratt string searching algorithm (or KMP algorithm) searches for occurrences of a “word” W within a main “text string” S by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters.

Edward is a fan of mathematics. He just learnt the Knuth-Morris-Pratt algorithm and decides to give the following problem a try:

Find the total number of occurrence of the strings “cat” and “dog” in a given string s .

As Edward is not familiar with the KMP algorithm, he turns to you for help. Can you help Edward to solve this problem?

Input

There are multiple test cases. The first line of input contains an integer T ($1 \leq T \leq 30$), indicating the number of test cases. For each test case:

The first line contains a string s ($1 \leq |s| \leq 1000$).

Output

For each case, you should output one integer, indicating the total number of occurrence of “cat” and “dog” in the string.

Example

standard input	standard output
7	4
catcatcatdoggy	1
docadosfascat	2
dogdddcats	5
catcatcatcatccat	3
dogdogdogdddoog	1
dcoagtcats	1
doogdog	

Note

For the first test case, there are 3 “cat” and 1 “dog” in the string, so the answer is 4.

For the second test case, there is only 1 “cat” and no “dog” in the string, so the answer is 1.