مسابقات برنامه‌نویسی دانشگاه فردوسی مشهد

FCPC

(طرح انتخابی)

ویژه دانشجویان فردوسی

۳۱ فروردین ۱۴۰۳

توضیحات:

- مسابقه شامل ۱۰ سوال در ۱۳ صفحه و زمان پاسخگویی ۴ ساعت است.
- کامپایلرهای موجود C, CPP, Java, Python هستند.
- ورودی‌ها و خروجی‌های تمام سوالات، ورودی/خروجی استاندارد است.

Question	Title
A	Fava or Java
B	Format Error
C	Delicious Food
D	Amin vs. Emad the Infinity War
E	Dragon-Warrior
F	Bull Fight
G	Sara and “Jo”
H	Mamali the painter
I	Negar and Niayesh
J	Mohzen and formula one

Problem A: Fava or Java

The classic programming language of FUM is Fava. They choose it instead of Java because it is simpler. This language is so peculiar and complicated. The language is that peculiar as it has exactly one variable, called *x*. Also, there are two operations:

- Operation ? ? increases the value of variable *x* by 1.
- Operation !! decreases the value of variable *x* by 1.

A statement in language Fava is a sequence, consisting of exactly one operation and one variable *x*. The statement is written without spaces, that is, it can only contain characters "?", "!", "X". Executing a statement means applying the operation it contains.

You're given a program in language Fava. The initial value of *x* is 0. Execute the program and find its final value (the value of the variable when this program is executed).

Input (standard input)

The first line contains a single integer $n(1 \leq n \leq 150)$,the number of statements in the program.

Next n lines contain a statement each. Each statement contains exactly one operation(? ? or !!) and exactly one variable *x*(denoted as letter “X”). Thus, there are no empty statement. The operation and the variable can be written in any order.

Output (standard output)

Print a single integer, the final value of x.

Sample Input and Output

Stdin	stdout
1 ??X	1
2 X?? !!X	0

Problem B: Format Error

The teacher has sent an e-mail to her students with the following task:

"Write a program that will determine and output the value of X if given the statement::

$$X = number_1{}^{power_1} + number_2{}^{power_2} + \cdots + number_n{}^{power_n}$$

And it holds that $number_1, \ldots, number_n$ are integers and $power_1, \ldots, power_n$ are one digit integers. Unfortunately, when the teacher downloaded the task to her computer, the text formatting was lost so the task transformed into a sum of N integers:

$$X = P_1 + P_2 + \cdots + P_n$$

For example, without text formatting the original task in the form of $X = 21^2 + 125^3$ became a task in the form of $X = 212 + 1253$. Help the teacher by writing a program that will, for given N integers from P_1 to P_n determine and output the value of X from the original task.

Input (standard input)

The first line of input contains the integer $N(1 \leq N \leq 10)$, the number of the addends from the task. Each of the following N lines contains the integer $P_i(10 \leq P_i \leq 9999, i = 1, \ldots, N)$ from the task.

Output (standard output)

The first and only line of output must contain the value of $X(X \leq 10^9)$ from the original task.

Sample Input and Output

stdin	stdout
2 212 1253	1953566
4 23 17 43 52 22	102
3 213 102 45	10385

Problem C: Delicious Food

Everyone's favorite character(haters will say it's fake) Erfan, has opened a new pizza place, the best in town. Erfan is trying to make the best pizza possible, but at the same time he doesn't want to have a small selection of pizzas.

He makes his pizzas out of N ingredients marked with numbers from 1 to N . All that would be simple if he could mix any ingredient with every ingredient on the pizza, but unfortunately, that is not the case. Sometimes some ingredients cannot mix and that creates additional complications for our pizza master.

There are M pairs of ingredients that cannot be on the same pizza at the same time. Given these restrictions, Erfan wants to know how many different pizzas he can make. Help him answer this question. Two pizzas are considered different if there is an ingredient of index i that is on one pizza, but not on the other.

Input (standard input)

The first line of input contains two integers N and M $(1 \leq N \leq 20, 1 \leq M \leq 400)$. Each of the following M line contains two different numbers a, b, they represent the prohibition of mixing ingredients marked with a and b on the pizza. $(1 \leq a, b \leq N)$. All pairs of ingredients are not necessarily distinct, some pair could occur multiple times.

Output (standard output)

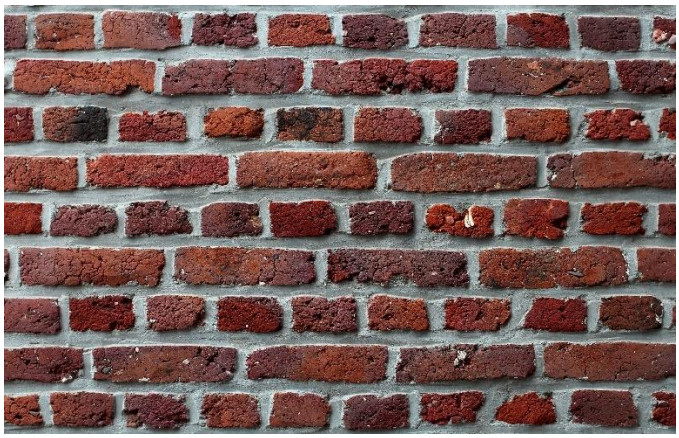
The first and only line of output must contain the number of different pizzas given the restrictions in the task.

Sample Input and Output

stdin	stdout
3 2 1 2 2 3	5
3 0	8
3 3 1 2 1 3 2 3	4

Problem D: Amin vs. Emad the Infinity War

In the previous contest, we learned that Amin is very angry with Emad, so he decided to lock him up in the faculty of engineering by placing a wall at the north entry and then closing all the other doors. Emad had no choice but to study math. While solving some problems, he became exhausted, so he decided to ask for help by posting a clarification on *judge.fcpc.ir*. Hamed, being a good teacher, checked the site as quickly as possible and aimed to enter the faculty to help Emad. Fortunately, Amin provided a way for other people to break through the wall. As you can guess, it was a math question.



"Find a path from the upper-left brick to the lower-right break using only right and down movements with the maximum product and shout out its reminder to $10^9 + 7$ to get the wall broken."

At that time Hamed gave up and didn't try to solve the problem but now you are here to help Emad so that he can at least get out of the faculty.

Input (standard input)

In the first line of input your program will be given n $(0 \leq n \leq 1000)$., the number of rows and columns of bricks in the wall.

The next n lines each contain n integers not less than 0 and not greater than 100.

Output (standard output)

In the only line of output print the answer .

Sample Input and Output

stdin	stdout
2 10 10 1 2	200
3 10 10 10 2 3 2 3 2 3	6000

Problem E: Dragon-Warrior

There are *N* cities and *M* roads in Dragons Country. Cities are numbered 1 through *N*, and each road connects exactly two cities. There are totally *K* dragons $D_1, D_2, \dots, D_k$ in these *N* cities. Dragon D_i lives in city C_i and has initially S_i heads. It grows N_i new heads every minute while alive! A dragon is alive if it has positive number of heads.

We want to hire a number of warriors to kill all dragons. We specify the initial city of each warrior. Then on each minute, each warrior can either go through a road from his current city to an adjacent city, or select an alive dragon in his current city and cut off one of its heads. We can specify the strategy of each warrior on each minute, and after they are done, any alive dragon D_i will grow its D_i new heads.

We want to find the minimum number of warriors with which all dragons can be killed in a finite amount of time.

Input (standard input)

There are multiple test cases in the input. For each test case, the first line contains three space-separated integers *N*, *M*, and *K*($1 \le N \le 300, 0 \le M \le \frac{N(N-1)}{2}, and\ 1 \le k \le 1000$). Each of the next *M* lines contains two integers *a* and *b*($1 \le a \ne b \le N$) indicating a road between cities a and b. The *i-th* line of the next *K* lines describes dragon D_i with three space-seperated integers $C_i, S_i\ and\ N_i (1 \le C_i \le N, 1 \le S_i \le 10^5, 0 \le N_i \le 10^5)$. The input terminates with a line containing 0 0 0 which should not be processed.

Output (standard output)

For each test case, output a line containing the minimum number of warriors who can eventually kill all dragons.

Sample Input and Output

stdin	stdout
2 1 1 1 2 1 7 4 4 4 2 1 2 2 4 4 1 1 3 1 2 3 2 3 1 0 0 0	5 2

Problem F: Bull Fight

Every year, Mamali brings his N cows to compete for ”best in show” at the state fair. His arch-rival, Khoras, brings his M cows to compete as well ($1 \leq N \leq 1000$, $1 \leq M \leq 1000$). Each of the $N+M$ cows at the event receive an individual integer score. However, the final competition this year will be determined based on teams of K cows ($1 \leq K \leq 10$), as follows: Mamali and Khoras both select teams of K of their respective cows to compete. The cows on these two teams are then paired off: the highest-scoring cow on Mamali’s team is paired with the highest-scoring cow on Khoras’s team, the second-highest-scoring cow on Mamali’s team is paired with the second-highest-scoring cow on Khoras’s team, and so on. Mamali wins if in each of these pairs, his cow has the higher score. Please help Mamali count the number of different ways he and Khoras can choose their teams such that Mamali will win the contest. That is, each distinct pair (set of K cows for Mamali, set of K cows for Khoras) where Mamali wins should be counted. Print your answer modulo 10^9+9.

Input (standard input)

The first line of input contains N, M, and K. The value of K will be no larger than N or M. The next line contains the N scores of Mamali’s cows. The final line contains the M scores of Khoras’s cows. Cow scores will not exceed 10^9.

Output (standard output)

Print the number of ways Mamali and Khoras can pick teams such that Mamali wins, modulo

$10^9+ 9$.

Sample Input and Output

stdin	stdout
10 10 3 1 2 2 6 6 7 8 9 14 17 1 3 8 10 10 16 16 18 19 19	382

Problem G: Sara and "Jo"

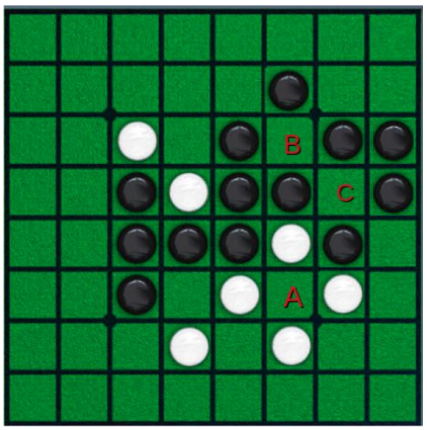
Sara likes the game "Jo" a lot. That's why she is looking for an AI to play "Jo" with. The Internet is disconnected, therefore she cannot just search google for a "Jo" AI and wants to make one herself.

"Jo" is played on an $n \times n$ table. Two players, one black and one white, play the game. In each state of the game, some cells are occupied by white beads while some others are occupied with black beads.

A path in the game table is defined as a sequence of cells in such a way that each two consecutive cells in the sequence are neighbors (two cells are neighbors if and only if they share a side).

Each empty cell is either a point cell for a player or a neutral cell. An empty cell *s* is a point for the white player if and only if every path between *s* and an arbitrary bead contains a cell with a white bead in it. Respectively an empty cell *s* is a point for the black player if and only if every path between *s* and an arbitrary bead contains a black bead.

The score of a player is defined as the number of empty cells that are points for that player. In order to make a good AI for "Jo", Sara needs to know the Score of each player in each state. As Sara is so lazy she told you to help her and write a program that given the current state of the game, determines the score of each player.



For example in the table above, cell "A" is a point for the white player while cells "B" and "C" are points for the black player and other cells are not point cells.

Input (standard input)

The first line of input contains a single number *n*, number of rows and columns of the game table.

The next *n* line make an $n \times n$ table. The character on the *i-th* row and *j-th* column is a_{ij}. a_{ij} can be either ".", "B", or "W" which represent an empty cell, Black bead cell, and white bead cell respectively.($1 \leq n \leq 1000$)

Output (standard output)

The only line of output should contain two integers. First the score of the black player and then the score of the white player.

Sample Input and Output

stdin	stdout
5 ..BW. ..BW. ..BW. ..BW.	10 5

Problem H: Mamali the painter

After a tough competition with Khoras, Mamali decided to change his job and become a painter. Since he was skilled at using computers, he chose to select some graphs and paint the shortest cycle of each, if one existed. The number of paint containers needed is the length of that cycle. As he is too busy painting the FCPC logo, you are asked to write the program to calculate how many paint containers Mamali needs.

You are given an array of n integers, $a_1, a_2, \ldots, a_n$ Consider a graph with n vertices $v_1, \ldots, v_n$ with an edge between v_i and v_j if and only if $a_i \; AND \; a_j \neq 0$ where AND is the bitwise and operation.

What is the amount of paint container that Mamali needs to paint the graph.

If there is nothing needed, print *-1*.

Note: A bitwise AND is a binary operation that takes two equal-length binary representations and performs the logical AND operation on each pair of the corresponding bits. Thus, if both bits in the compared position are 1, the bit in the resulting binary representation is 1 (1 × 1 = 1); otherwise, the result is 0 (1 × 0 = 0 and 0 × 0 = 0). For example:

0101(decimal 5) AND 0011(decimal 3) = 0001(decimal 1)

Input (Standard Input)

the first line contains one integer *n*($1 \leq n \leq 10^5$).

The second line contains *n* integers $a_1, a_2, \ldots, a_n (0 \leq a_i \leq 2^{60})$

Output (standard output)

If the graph doesn't have any cycles, output -1, otherwise, print the length of the shortest cycle to help Mamali.

Sample Input and Output

stdin	stdout
4 3 6 28 9	4
5 5 12 9 16 48	3
4 1 2 4 8	-1

Problem I: Negar and Niayesh

Negar and Niayesh are playing a board game.

On the board, there are *N* cells numbered 1 through *N* and, *N-1* roads, each connecting two cells. Cell a_i is adjacent to cell b_i through the *i-th* road. Every cell can be reached from every other cell by repeatedly traveling to an adjacent cell.

Initially, cell 1 is painted black, and Cell *N* is painted white. The other cells are not yet colored. Negar(who goes first) and Niayesh (who goes second) alternately paint an uncolored cell. More specifically, each player performs the following action in her turn:

- Negar: selects an uncolored cell that is adjacent to a black cell, and paints it black.
- Niayesh: selects an uncolored cell that is adjacent to a white cell, and paints it white.

A player loses when she cannot paint a cell. Determine the winner of the game when Negar and Niayesh play optimally.

Input (standard input)

First line contains N. *i-th* line of the next N lines contains a_i and b_i.

$(2 \leq N \leq 2.10^5, 1 \leq a_i, b_i \leq N)$

Output (standard output)

If Negar wins, print “Negar”; if Niayesh wins, print “Niayesh”.

Sample Input and Output

stdin	stdout
7 3 6 1 2 3 1 7 4 5 7 1 4	Negar
4 1 4 4 2 2 3	Niayesh

Problem J: Mohzen and formula one

Mohzen is an avid Formula 1 enthusiast. For that reason he has several questions from this competition. Due to time constraints, we will only be able to check two of them.

First question: Does formula 1 cars have clutch?

Second question: There are N cars racing along an infinite track. The *i-th* car starts its race in position X_i and rides with speed V_i along the track for exactly T seconds. The race track should be divided into lanes so that the cars wouldn't crash into each other. In particular, no two cars in the same lane can be in the same position. No cars are allowed to change lanes, speed or start point. What's the least number of lanes needed to have a fair and accident–free race?

Since Mohzen is too lazy, he is only capable of answering one question, and as you can guess, it will be the first one. Therefore, you have to answer the second one.

Input (standard input)

The first line of input contains N and $T(1 \leq N \leq 10^5, 1 \leq T \leq 10^9)$.

Next comes N lines, each containing two integers, x_i *and* $v_i(0 \leq x_i \leq 10^9, 1 \leq v_i \leq 10^9)$. It's guaranteed that all x_i's are distinct.

Output (standard output)

In a single line, print the minimum number of lanes necessary.

Sample Input and Output

stdin	stdout
5 3 0 1 1 2 2 3 3 2 6 1	3