مسابقات برنامه‌نویسی دانشگاه فردوسی مشهد

FCPC

(طرح آموزشی)

ویژه دانشجویان فردوسی

۱۹ اسفند ۱۴۰۲

توضیحات:

- مسابقه شامل ۸ سوال در ۱۱ صفحه و زمان پاسخگویی ۳ ساعت است.
- کامپایلرهای موجود C, CPP, Java, Python هستند.
- ورودی‌ها و خروجی‌های تمام سوالات، ورودی/خروجی استاندارد است.

Question	Title
A	Grade Tax
B	Key Maker
C	Valid Emails
D	FCPC Logo
E	ENTEKHAB VAHED
F	Password
G	The Big Surprise
H	Lets Burn and Rob Manhootan

Problem A: Grade Tax

The new grading system of FUM is very precise. The grade scale is from 0 to 10,000,000 instead of 0 to 20. And also they have come up with a new concept called the Grade Tax in order to reduce the total average of grades, which works as follows:

- For the initial grades less than or equal to 1,000,000 no Grade Tax is considered.
- For the initial grades greater than 1,000,000 and less than or equal to 5,000,000 the Grade Tax is 10% of the initial grade.
- For the initial grades greater than 5,000,000 the Grade Tax is 20% of the initial grade.

The final grade for each student is their initial grades mines their Grade Tax. You should write a program to calculate the final grade for any given student.

Input (standard input)

There are multiple lines in the input. Each line contains a student's initial grade, which is a positive integer, a multiple of 1000 (and not greater than 10,000,000). The input terminates with a line containing 0 which should not be processed.

Output (standard output)

For each student, output a line containing the final grade.

Sample Input and Output

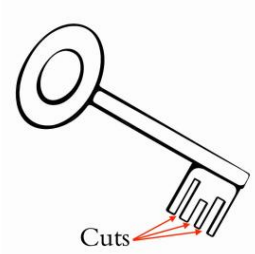
Standard Input	Standard Output
10000 50000 2000000 7500000 0	10000 50000 1800000 6000000

Problem B: Key Maker

Hassan is a happy key maker. Every customer arrives with a safe-box key, and asks him to create some copies of the key. Each key has several cuts of different depths. The picture below shows a safe-box key with 3 cuts. To make a copy, Hassan needs to make the same number of cuts with exactly the same sequence of depths in a new blank key.

In the first days of his job, Hassan wasted many blank keys to make copies. Most of the copied keys, however, did not match the customer keys and he could not sell them. He collected those copied keys in a trash-box, and now he is thinking of recycling them.

When a new customer arrives, Hassan looks into the trash-box, collects all keys with the same number of cuts as the customer's key, and counts the keys that can match the customer's key. A key can match the customer's key if it already has exactly the same sequence of cut depths, or the depth of some of its cuts can be increased to reach the same sequence. Since this job is too hard for him, he has asked your help. For simplicity, you can assume that in any two keys with the same number of cuts, the position of the cuts along the keys are identical.



Input (standard input)

There are multiple test cases in the input. The first line of each test case contains two space-separated integers m as the number of cuts in the customer's key ($1 \leqslant m \leqslant 10$), and n as the number of keys with the same number of cuts in the trash-box ($1 \leqslant n \leqslant 100$). The second line of the test case consists of m space-separated integers, as the depths of cuts in the customer's key. Each of the next n lines also contains m integers, as the depths of cuts in a trash-box key. The depth of cuts in each of these n + 1 keys are 1-digit positive integers given in the left-to-right order. The input terminates with a line containing 0 0 which should not be processed.

Output (standard output)

For each test case, print a single line containing the number of keys in the trash-box that either match the customer's key or can be cut to match it.

Sample Input and Output

Standard Input	Standard Output
4 1 3 2 1 3 2 2 1 2 4 1 4 2 2 2 3 2 2 3 5 3 2 2 4 2 2 2 3 4 3 2 1 1 3 2 2 2 2 2 2 2 0 0	1 0 2

Problem C: Valid Emails

For this FCPC contest, many people registered with several email addresses. The person responsible for the registration wanted to see how many valid and distinct email addresses registered. However, since she is busy trying to blow up the balloons for the contest, she asked you to find out the number of distinct email addresses that are valid.

A valid email address consists of a username and a domain name separated by a character '@'. A username is a string containing letters (a-z and A-Z), digits (0-9), underscores (_), and periods (.). Usernames cannot begin or end with a period and cannot contain two consecutive periods. Other than this rule, periods do not matter in email addresses (they can be removed without changing the address). Uppercase and lowercase letters in the usernames are considered the same. So, usernames AliBaba and ali.baba are considered the same. Usernames should contain 6 to 30 characters, after removing all of its periods.

A valid domain name is a string of length between 3 and 30 (inclusive), consisting of domain parts separated by periods (.). A domain name must not start or end with a period. Each domain part is a nonempty string of letters (a-z and A-Z), digits (0-9), and dash (-). Uppercase and lowercase letters in the domain names are also considered the same. So, Foo.bar is the same as foo.Bar, but not the same as Foo-Bar or Foobar.

Input (standard input)

The first line of the input contains a positive integer n ($1 \leqslant n \leqslant 1000$), the number of the registered email addresses. Each of the next n lines contains one email address of length at most 100 and consisting of alphabets, digits, '@', '.', '_', and '-'.

Output (standard input)

Print a single integer which is the number of distinct email addresses that are valid

Sample Input and Output

Standard Input	Standard Output
5 acmacm@icpc.ir acmacm..a@icpc.ir .a1a1.abc@icpc.ir acma.c.m@icpc.ir acmacm@icpc.com	2

Problem D: FCPC Logo

FCPC contest is finally going to be held after 5 years. In order to design the logo, the organizing-committee decided to publicly call for logos. It was not surprising that many logos were received in a short time as the students were so passionate for the contest. In the first round, logos were judged by some professional graphic designers, and the best logos being artistically capable to be the FCPC logo were selected to be judged in the second round.

The selected logos are now presented to the organizing-committee members for voting. The voting systemis a little bit complicated: each member can vote for at most three different logos in some order. The first, second and third choices of each member are awarded 3, 2 and 1 points, respectively. The score of a logo is the total points the logo receives from all members. The logo with the highest score is the winner. In the case of ties, the winner is the logo with higher number of first votes. Again, if some logos have the same score and first votes, the logo with more second votes is the winner. If we still have ties, all of them would be winners. Given the voting information, your job is to identify the winner logo (or logos).

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains a positive integer n denoting the number of voters ($1 \leqslant n \leqslant 100$). Each of the next n lines starts with an integer d_i, representing the number of logos chosen by the *i-th* voter ($1 \leqslant d_i \leqslant 3$), followed by d_i different logo IDs showing the choices of that voter (from left to right). Each logo ID is a positive integer not exceeding 10^6. All integers in a line are separated with a single space. The input terminates with a line containing 0 which should not be processed.

Output (standard output)

For each test case, output a line containing the winner logos in the increasing order of their IDs. Logo IDs in a line must be separated with a single space.

Sample Input and Output

Standard Input	Standard Output
4 3 5 2 1 3 12 5 2 2 1 2 3 2 1 5 2 3 3 2 1 3 2 3 1 0	2 2 3

Problem E: ENTEKHAB VAHED

"ENTEKHAB VAHED" time has arrived again. Saman who is tired of life since he is been getting rejected in his driving license exams for the last couple of years, has decided to have the best "ENTEKHAB VAHED" through his educational experience. So he asked his professors for the best course order in their opinion.

Saman has k professors, each one ranks courses from a specific point of view. There are n courses numbered 1 through n, and each professor suggests one permutation of these n courses.

In order to have the best "ENTEKHAB VAHED", Saman has to find a permutation in which, if i is ranked before j, then at least half of the professors should have ranked i before j.

Since Saman is too depressed to do anything, he wants your help in finding the best permutation if there exists one.

Input (standard input)

There are multiple test cases in the input. The first line of each test contains two space-separated integers n ($1 \leq n \leq 1000$) and k ($1 \leq k \leq 200$), the number of courses and the number of the professors, respectively. Then there are k lines, where the *i-th* line contains a permutation of n numbers, from 1 to n, representing the suggested permutation of the *i-th* professors. The input terminates with a line containing 0 0 which should not be processed.

Output (standard output)

For each test case, print a single line containing a permutation of n numbers, from 1 to n, representing the best possible "ENTEKHAB VAHED".

If there are more than one answer, print the first one in lexicographic order (a sequence a_1 to a_n is lexicographically less than a sequence b_1 to b_n if there exists a positive integer j such that $a_i = b_i$ for all $1 \leqslant i \leqslant j - 1$ and $a_j < b_j$) . If no such "ENTEKHAB VAHED" exists, write "No solution" instead.

Sample Input and Output

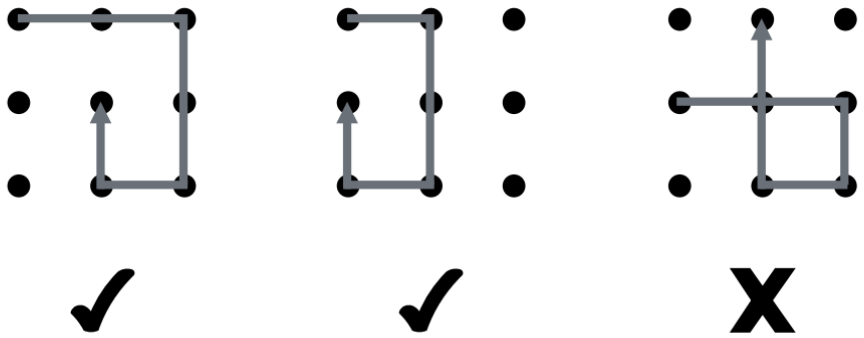
Standard Input	Standard Output
5 3 3 2 4 1 5 4 1 5 2 3 2 4 5 1 3 5 2 5 4 3 2 1 1 2 3 4 5 4 3 1 4 2 3 4 2 3 1 3 1 2 4 0 0	2 4 1 5 3 1 2 3 4 5 No solution

Problem F: Password

After working for several months at Part Software Group, Farhad became rich enough to buy a house in the valley of the rich. There he met Shirin several times. Now, he is considering proposing to her whether she would marry him. To surprise her, he wants to install an application on her phone that pops up at the exact right time and asks if she would marry him.

However, to install the application secretly, he needs her password which he unfortunately does not have. He knows her password is a poly-line consisting of vertical or horizontal line segments. Each line segment connects the center of two cells in a 3 × 3 grid. Looking at her hand while she unlocked her phone, Farhad learned the direction of each line segment. However, he was too distracted to also learn the length of each segment. He also knows that her phone's operating system does not allow the poly-line to intersect with itself even in one point.

Farhad wants to distract Shirin long enough to try all possible patterns given what he already knows. Unfortunately, he has no idea how long that will take. That is why, he has now turned to you for help. Help him by writing a program that calculates the total number of possible password patterns given the direction of the line segments. The following figure depicts two valid and one invalid patterns given the line segments were directed towards right, down, left, and up in order.



Input (standard input)

In the only line of the input, a single string is given consisting of characters R, U, L, and D which represent a line segment toward right, up, left, and down, respectively. The length of this string is at most 10. Every two consecutive characters is guaranteed to be different

Output (standard output)

In the only line of the output, print the number of patterns satisfying Farhad's knowledge of the password. Note that this number might be zero.

Sample Input and Output

Standard Input	Standard Output
DRU	15

Standard Input	Standard Output
R	9

Problem G: The Big Surprise

"A big surprise is coming on the next Thursday!", the young mayor of TetrisCity announced in the social media. TetrisCity is the most populous and modern city in Neverland constructed on a flat area with endless clusters of high-rise buildings packed so closely together that they resemble a game of Tetris. Buildings look like axis-parallel boxes constructed on the ground and they are disjoint (they do not even touch each other).

The big surprise announced by the mayor is going to be a special delivery service using drones. The drones used in this service are a generation of quadcopters which can physically move only in one of x, y, and z directions. So, the distance traveled by a drone is the sum of distances traveled by it in each axis. The young mayor now has ordered to make the drones smart by equipping them with a software that computes the shortest path from any source to any destination avoiding the buildings. Your job is to develop this software.

Input (standard input)

The first line of the input contains an integer n ($0 \leqslant n \leqslant 100$), specifying the number of buildings in the TetrisCity. Each of the next n lines contains 5 space-separated integers x, y, x', y', and h specifying a building: the coordinates (x, y) and (x', y') respectively specify the west-south corner and the east-north corner of the building, and h determines its height. It is guaranteed that the volume of the building is not zero. The source and destination appear at the end of the input in two separated lines; each containing x, y, and z coordinates. All numbers in the input are non-negative integers being at most 100 000. It is guaranteed that the source and destination are outside the buildings (they can be on the boundary of buildings). The shortest path can touch buildings and it is assumed that a drone looks like a point.

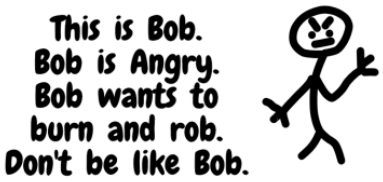
Output (standard output)

In the output, print the length of the shortest path from the source to the destination avoiding the buildings.

Sample Input and Output

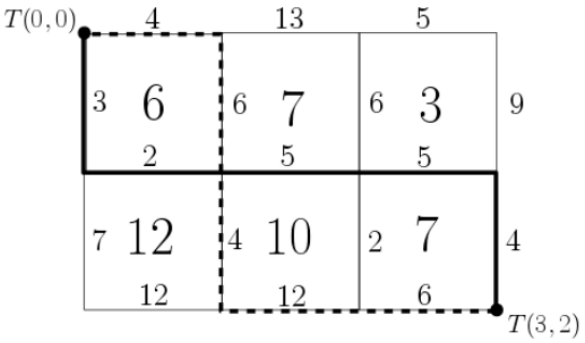
Standard Input	Standard Output
1 1 1 11 21 40 5 0 5 6 23 8	35

Problem H: Lets Burn and Rob Manhootan



There are two types of angry people in this world, those who burn and those who rob. But we programmers know that there is a third type; those who counter their anger, by both burning and robbing.

Bob lives in Manhootan. The city of Manhootan is like a grid of n rows and m columns, containing $n \times m$ blocks. The rows are numbered from 0 to $n-1$ from north to south and the columns are numbered from 0 to $m-1$ from west to east. The j-th block on i-th row is worth A_{ij}. Before the first row, between every two consecutive rows, and after the last row, there is a west-east street. The $n+1$ west-east streets are numbered from 0 to n from north to south. Similarly, before the first column, between every two consecutive columns, and after the last column, there is a north-south street. The $m+1$ north-south streets are numbered from 0 to m from west to east. The part of a street that is between two adjacent blocks is called a *street segment*. Each west-east street contains m street segments, numbered from 0 to $m-1$ from west to east. Similarly, each north-south street contains n street segments, numbered from 0 to $n-1$ from north to south. Since Manhootan is an expensive city, passing through street segments costs money. Passing through the j-th segment of the i-th west-east street costs H_{ij} and passing through the j-th segment of the i-th north-south street costs V_{ij}.



After a recent crisis in Manhootan, Bob got angry. He pierced his car's fuel tank to make it leak on the streets he passes. Let's call the intersection of i-th west-east street and j-th north-south street, $T(i,j)$. At first, Bob is at $T(0,0)$. He is planning to drive to $T(n,m)$ only going east and south, then returning to $T(0,0)$ only going west and north. Then, he is going to light the leaked fuels and put the streets on fire. After that, Bob will rob all the blocks that are caught inside the fire, i.e., any block that can not reach outside of Manhootan without crossing a burning street, will be robbed by Bob. The figure shows one possible plan for Rob in the sample.

Now, you can't be like Bob, but you can help him find the most profitable burn-and-rob plan. In other words, maximize the total value of the robbed blocks minus the total cost of the passed street segments. A street segment may be passed twice, which should be paid for each separately.

Input (standard input)

The first line of input contains two integers n and m ($1 \leqslant n, m \leqslant 200$), the number of rows and columns, respectively. The next n lines describe the value of blocks; each containing m numbers, where the j-th number of the i-th line denotes A_{ij} ($1 \leqslant A_{ij} \leqslant 100$). The next $n+1$ lines describe the cost of west-east street segments. Each line contain m numbers, where the j-th number of the i-th line denotes H_{ij} ($1 \leqslant H_{ij} \leqslant 1000$). Finally, the next $m+1$ lines describe the cost of north-south street segments. Each line contains n numbers, where the j-th number of the i-th line denotes V_{ij} ($1 \leqslant V_{ij} \leqslant 1000$).

Output (standard output)

Print the profit of the most profitable plan. Note that the answer can be negative, zero, or positive.

Sample Input and Output

Standard Input	Standard Output
2 3 6 7 3 12 10 7 4 13 5 2 5 5 12 12 6 3 7 6 4 6 2 9 4	-28