

Intra IUB Programming Contest Spring 2026

Problemset (6 Problems)

Saturday, April 11, 2026

Problem A – A Tale of Two Merchants

In the lively bazaar, BiqarBoy proudly displays his stock—each item marked by a lowercase letter. Vutu, never one to back down, makes a bold claim: “Anything BiqarBoy can offer, I can match exactly!” The townsfolk, curious to see if Vutu’s boast holds true, devise a test. If Vutu’s list of letters can be rearranged to perfectly mirror BiqarBoy’s, then his claim stands. Otherwise, his words are just empty chatter. Given the two lists of letters as strings, determine if Vutu’s claim holds true.

Input

First line of the input is a single integer T ($1 \leq T \leq 100$) denoting the number of test cases. This is followed by $2T$ lines, two for each test case, each line containing a single string with maximum length 100 made up of lowercase English letters.

Output

For each test case, the output should contain a single line with either “YES” or “NO” (without quotes), indicating whether Vutu’s claim is substantial or not.

SAMPLE INPUT	SAMPLE OUTPUT
6	YES
silent	NO
listen	YES
there	YES
their	NO
dormitory	NO
dirtyroom	
elevenplustwo	
twelveplusone	
seal	
lease	
satires	
tastier	

Problem B – Balanced Duels

During a big gaming event, two famous players Biqarboy and Fox Ciel finally faced each other in a popular multiplayer game. They played N matches consecutively, with each match ending with one of the two as the victor. The entire sequence was recorded as well.

The event organizers want to upload a single video of consecutive matches online. However, although the two players are very respectful to each other, their massive online fanbases may not necessarily be as civil, and it is possible that if the uploaded video showed one player winning more matches than the other, this may result in some unpleasant reactions.

Therefore, it was decided that the sequence of matches to upload should have both players winning the same number of matches. The sequence should be consecutive (i.e., a continuous recording with no cuts) and should be as long as possible.

Help the event organizers determine the longest number of consecutive matches where each player has an equal number of victories.

Input

First line of the input is a single integer T ($1 \leq T \leq 20$) denoting the number of test cases. This is followed by T lines, one for each test case, each containing a string of length from 1 to 100000. Each character is either B or F, indicating whether Biqarboy or Fox Ciel won the match respectively.

Output

For each test case, the output should contain a line with a single integer, denoting the maximum length of a consecutive sequence where Biqarboy and Fox Ciel won an equal number of matches

SAMPLE INPUT	SAMPLE OUTPUT
3	4
BBFBFBFB	0
BBBBBBBBBBBB	12
FBFFFFBFBFBFFFFB	

Problem C – Course Outcomes

Undergraduate courses are typically designed to achieve certain programme outcomes. Not all courses contribute equally to each outcome; for example, some courses may focus more on theory while others focus on problem-solving. Each course is assigned a weight factor for each programme outcome, so a student's progress towards an outcome can be estimated by multiplying their course marks with their respective weight factors, and adding the results.

You are given two matrices A and B , where A contains the marks obtained by students (rows) in courses (columns), while B contains the weight factors of the courses (rows) towards the programme outcomes (columns). In order to determine the students' progress towards the course outcomes, we need to multiply matrix A with matrix B .

Multiplying two matrices is only valid if the number of columns in the first matrix is equal to the number of rows in the second matrix. When multiplying $A \times B$, the elements of the resulting matrix C are determined by the following formula:

$$C_{i,j} = \sum_{k=1}^L (A_{i,k} \times B_{k,j})$$

where $C_{i,j}$ denotes row i and column j of matrix C , and L denotes the number of columns in A and the number of rows in B . Given matrix A with N rows and M columns, and matrix B with P rows and Q columns, give the result from multiplying $A \times B$, or indicate this is invalid.

Input

First line of the input is a single integer T ($1 \leq T \leq 100$) denoting the number of test cases. Each test case begins with a line containing four integers N, M, P, Q ($1 \leq N, M, P, Q \leq 20$), separated by a space, denoting the dimensions of the two matrices. The next N lines each contain M integers, representing the first matrix. Finally, the next P lines each contain Q integers, representing the second matrix. All matrix elements range from 0 to 100.

Output

For each test case, if the two matrices cannot be multiplied in the given order $A \times B$, then the output should contain a single line with "Invalid" (without quotes). Otherwise, the output should indicate the elements of the output matrix, with each row in a separate line, and elements within the same row are to be separated by a single space.

SAMPLE INPUT	SAMPLE OUTPUT
2	59 75 36
3 2 2 3	54 27 18
4 7	46 75 34
9 0	Invalid
1 8	
6 3 2	
5 9 4	
2 3 2 4	
7 1 4	
3 7 2	
8 3 9 1	
2 5 0 6	

Problem D – Dark Streets

In a certain area, there are horizontal and vertical streets that form a perfect grid. Some of the intersections have streetlights, and there are no other lights except at the intersections.

Recently, in order to conserve energy, the buildings are kept out of power outside of working hours. It was reported that the area can become very dark late at night, with the streetlights being the only source of illumination.

For safety purposes, a decision was made that every row in this grid should have at least one streetlight and every column should have at least one streetlight. The task now is to determine the minimum number of new streetlights that need to be installed in order to achieve this. Note that the streetlights that are currently present cannot be moved.

Input

First line of the input is a single integer T ($1 \leq T \leq 100$) denoting the number of test cases. Each test case begins with a line containing four integers R, C ($1 \leq R, C \leq 100$), separated by a space, denoting the number of rows and the number of columns respectively in the street grid. The next R lines each contain a string of length C , where each character is either 0 or 1, where 1 indicates that there is a light at this intersection, while 0 indicates that there is no light at this intersection.

Output

For each test case, the output should contain a line with a single integer, denoting the minimum number of new streetlights that need to be installed.

SAMPLE INPUT	SAMPLE OUTPUT
3	1
3 3	9
100	0
000	
010	
2 10	
10000000000	
10000000000	
5 5	
00100	
10000	
00001	
00010	
01000	

Problem E – Exam Admittance

IUB has recently begun to enforce a policy that requires students to attend 60% of the classes in a given course in order to be eligible to take the final exam for that course.

You are asked to construct a program that can read the number of classes a student was present in and the number of classes a student was absent in and determine whether the student should be eligible to take the final exam or not.

In order to avoid any issues of having an outdated system for future developments, you should ensure that your program can correctly handle even very large numbers (up to 10^{18}).

Input

First line of the input is a single integer T ($1 \leq T \leq 100$) denoting the number of test cases. This is followed by T lines, one for each test case, each containing two integers P and A ($0 \leq P, A \leq 10^{18}, P + A > 0$) separated by a space, indicating the number of classes marked present and absent respectively.

Output

For each test case, the output should contain a single line either “YES” or “NO” (without quotes), indicating whether the student should be admitted or not.

SAMPLE INPUT	SAMPLE OUTPUT
6	YES
60 40	NO
151 300	NO
812 812	YES
22222222222 0	NO
900 1500	NO
599999999999999999 400000000000000001	

Problem F – Fuel Shortage

Due to certain circumstances, there is a shortage of various resources, particularly oil and natural gas, in many parts of the world. This includes the various types of fuel needed by cars and similar vehicles. As a result, refuelling stations (sometimes referred to as “gas stations”) are now restricting how much fuel they can provide to each customer.

Vutu is normally very busy, but he now finally has a day off, allowing him to refuel his car. For simplicity, we describe all fuel quantities based on the distance that they allow Vutu to cover with his car.

Vutu's car currently has enough fuel to drive S kilometers. There are N refuelling stations that Vutu can go to. Each station is a distance D kilometers from Vutu's home, and can provide fuel that is equivalent to R kilometers in Vutu's car. However, the total fuel in Vutu's car can never exceed the maximum capacity, which is equivalent to M kilometers in this car.

Due to the long queues in these refuelling stations, Vutu is only able to visit at most one station today. He would need to travel with his car from his home to the station, refuel, and then return home from the station with his car. Determine the maximum amount of fuel that his car can have at the end of the day.

Input

First line of the input is a single integer T ($1 \leq T \leq 100$) denoting the number of test cases. Each test case begins with a line containing four integers S, M, N , separated by a space, denoting the starting fuel (in kilometers), maximum capacity (in kilometers) and number of stations. The next N lines each contains two integers D and R , separated by a space, denoting the distance (in kilometers) from the station to Vutu's home and the quantity of fuel (in kilometers) they can provide respectively. All integers range from 1 to 1000 and $M > S$.

Output

For each test case, the output should contain a line with a single integer, denoting the maximum possible fuel quantity (in kilometres) that Vutu's car can have at the end of the day.

Sample Input	Sample Output
2	70
60 100 3	30
60 90	
10 30	
35 500	
30 1000 2	
32 500	
10 15	

In the first test case, there are three stations. For the first station, the fuel changes are $60 - 60$ (travel) + 90 (refuel) - 60 (return) = 30 . For the second station, the changes are $60 - 10$ (travel) + 30 (refuel) - 10 (return) = 70 . For the third station, the changes are $60 - 35$ (travel), but the refuel sets the car to the capacity of 100, before the return trip reduces it to $100 - 35 = 65$, so the best outcome is 70 from the second station.

In the second test case, Vutu's car cannot make it to the first station, but going to the second station results in a net loss of fuel, so it's best to keep the car at home without refuelling.