Problem Set

The 2022 ICPC Asia Topi Online Preliminary Programming Contest

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- Do not create disturbance or move around unnecessarily in the arena.
- If you have any question regarding the problems, send a clarification from the judges using DOMJudge.
- There would be no internet access and mobile phones are also not allowed.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3.
- Source code file name should not contain white space or special characters.
- You must take input from Console, i.e., Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console, i.e., Standard Output Stream (stdout in C, cout in C++, System.out in Java).
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity, etc.**
- **Pay special attention while taking input. Some languages may include carriage return as an extra line.**
- Unless mentioned in some problem, all your programs must meet the time constraint of 5 seconds.
- The decision of judges will be final.
- **The link for DomJudge is:** https://domjudge.giki.edu.pk:8022/domjudge/public

PROBLEM 1: Tricky Path

Time limit: 5 seconds

The city of Dubai has *N* districts numbered from 1 to *N*. Roads are built in such a way that each pair of neighborhoods is connected. District *A* and *B* can be considered as connected if and only if there is a road between district *A* and district *B*, or there is a district *C* such district *A* is connected to *B* and *B* and *C*, and so on.

You are a salesperson that needs to plan visits to all districts using minimum distance. However, you soon realize that all other salespersons also have the same idea resulting in huge traffic jams. Therefore, you decide that to save time, you will choose the path that is just one step higher than the minimum distance to cover all the districts and that will ultimately save you time. You are, therefore, required to compute such path and the total distance you need to travel. Note that you do not have to return to the starting district.

Input

The first line in the input file is the number of test cases. The next line is an integer *N* (3<=N<=100) representing the number of districts for that case. This is followed by *N* lines each containing *N* integers. An integer in the *i*-th line and *j*-th place represents the distance [1,1000] between district *i* and district *j*. Any value greater than 1000 implies the absence of a direct link between the two districts.

Output

Your output should have as many lines as the number of test cases. Each line is the length of the distance to be travelled such that all districts are visited.

Sample Input	Sample Output
2 3 0 990 692 990 0 179 692 179 0 6 0 13 9999 9999 39 27 13 0 7 9999 9999 28 9999 7 0 7 9999 2 9999 9999 7 0 36 14 39 9999 9999 36 0 34 27 28 2 14 34 0	1169 65

PROBLEM 2: The closely linked mammals – Dolphins and Humans

Time limit: 3 seconds

Dolphins are sea animals, which are classified as mammals and not fish. Dolphins are warm blooded like all mammals do. Dolphins use their lungs to breathe air, as opposed to fish, who do it through their gills. Dolphins must frequently go to the water's surface to breathe. It is simple for a dolphin to surface for air because of the blowhole on top of its head, which serves as a "nose". Dolphins are social and intelligent animals and have a close link with human beings in this regard.

Due to such well-developed brain structure (like humans), and recent popularity of neural networks and deep learning architecture, scientists are studying the proteins of dolphins in close proximity to humans. A unique characteristic of dolphin's genome has amazed the scientists – instead of 20 amino acids, it contains 26 amino acids, all represented by alphabets in capital letters. The sequence (and identity) of dolphin's amino acid sequences matters but some of their amino acids can morph into other amino acids at will but retain the function of the original amino acid. The scientists were able to identify the equivalent amino acids between humans and dolphins through extensive experimentation.

You are provided a list of proteins that might or might not be substitutes, as well as a list of equivalent amino acids. You must check to see if the proteins in each pair are equivalent. If two proteins have the same length and each of the first protein's amino acids can be converted into the corresponding amino acid of the second protein using the available equivalence rule at least once, they are said to be homologous.

For example, in the first test case (fourth example) given below, the strings are FFO and SHT. When finding equivalence between both strings, F translates to S directly (F→S), F translates to H via S (F→S→H), and finally O translates to T via C (O→C→T). Hence, FFO is equivalent to SHT, and the output is Yes.

Input

The first line in the input file is the number of test cases. Each test case is organized as follows:

1. The first two lines of each test case represent the mapping. The first of these lines contain space-separated amino acid letters (Y_1 Y_2 Y_3 … Y_n) and the second line contains the equivalent of each amino acid of line 1 separated by space (Z_1 Z_2 Z_3 … Z_n).
2. The next line represents the number of strings to match, *N*.
3. Each subsequent *N* lines contain the pair of strings to match, separated by a space.

Output

Your output should have as many lines as the number of strings to match (of all test cases). For each pair of strings, display Yes if the two strings are equivalent and valid, and No otherwise.

Note: Amino acids and protein pairs use only uppercase letters 'A'–'Z', and each protein sequence contains **at least 3** and **at most 20 letters**.

Sample Input	Sample Output
2 C I K O R T T S W F T R P C O E F H P S 5	Yes No No Yes

KROC KROC NAC NOC STTKCAR STEKCOR FFO SHT SKOOITSF SKCOREHS A B A C C A B A 6 AAA ABC ABCD AAAA ACM BCM AC BC ACABAACCBBCCAA BBBACCBBCCABBC ABACABACABACABACABACABACA ABACABACABACABACABACABACA	Yes Yes No Yes No Yes No

PROBLEM 3: Emergency travelling plan

Time limit: 5 seconds

A traveler is visiting different places and suddenly receives a call from the office to reach back office due to an emergency meeting. Now the traveler is required to move back as soon as possible but he has certain constraints. There are some cities that are connected through road networks, and some are also connected through air connections. The traveler may not stop in a city where there is an aerial route taking him back to previous visited cities. As he is in far remote areas and the roads are not that good so he can travel to a maximum of 6 cities in one day if he travels by road. Once he reaches the city having the aerial route, he reaches the other destination of the aerial route by the end of the same day.

Let us take an example given in the figure given below. In figure all cities are connected through road network shown with arrow. There are some cities that also have aerial route shown as dotted lines such as from city 2 to city 21. Consider the traveler wants to move from city 1 to city 30 in as minimum days as possible. The traveler can travel up to six cities by road, but there is an aerial route available at city 2 so take one day to travel to city 2, take a flight to city 21 on day 1. After that he can take the road, but he should not stop at city 26 as he does not want to fly back (In case of such case the traveler should either stay on previous city or next city in case if maximum 6 cities are not covered). So, on day 2, he reaches city 27, covering 6 cities by road and on 3rd day he reaches his destination i.e., city 30. (Same can be achieved by travelling from city 1 to city 7 then on second day reaching city 25 through city 10 and on third day reaching destination). So, in total the minimum days that is required to reach destination is 3 days.

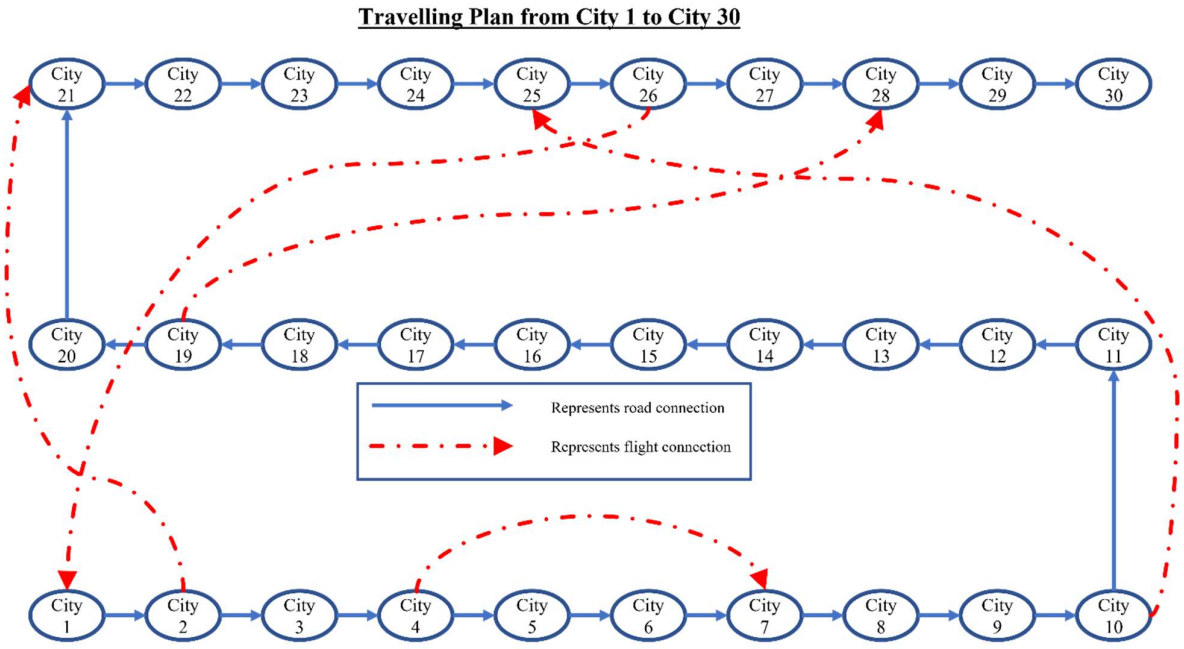


Figure: Emergency Travel plan

Hints:

- Start number of cities with zero i.e., city 0 and last city will be city (N-1) where is N is the number of cities.

Input

The first line should contain the number of cases. For each test case, the starting line contains the number of cities, the next line shows the number of aerial routes (*n*), and the subsequent *n* lines will show the source and destination of the aerial route, separated by a space.

Output

There would be as many outputs as the number of test cases. The output displays the minimum number of days required to travel from city 1 to city N.

Sample input	Sample Output
2 30 5 2 21 4 7 10 25 19 28 26 0 98 3 5 29 35 23 24 95	3 4

PROBLEM 4: Adjacent tiles

Time limit: 5 seconds

You are helping an interior designer to design a layout of a floor using square tiles. The interior designer is interested to form a tile pattern that can be verified using a mathematical formula. The formula needs the number of adjacent tiles of white color as an input. The pattern consists of only two colors, i.e., Black and White.

You are required to help the designer by calculating the number of adjacent white tiles using a program. The floor layout can be represented as a grid. Given the coordinates of a grid cell that contains white tile, what is the area of the white pattern containing that cell (an area is measured by number of grid cells it contains. Diagonal cells are considered to be adjacent).

Input

The input begins with a single positive integer indicating the number of the cases. This is followed by a line consisting of two integers D and Q, where D denotes the dimension of the floor (floor has always square dimension) and Q denotes the number of queries.

This is followed by D lines representing $D \times D$ grid covering the floor where a 'W'/'B' at the j-th character of the i-th line indicates black/white tile within the cell at row i and column j of the grid.

The pairs of integers in the last query consists of the coordinates (i, j) of the grid, $0 < i \leq D$,$0 < j \leq D$.

The pairs of integers on the last Q lines of the test case each representing the row and column numbers of some grid cell that contains white tile.

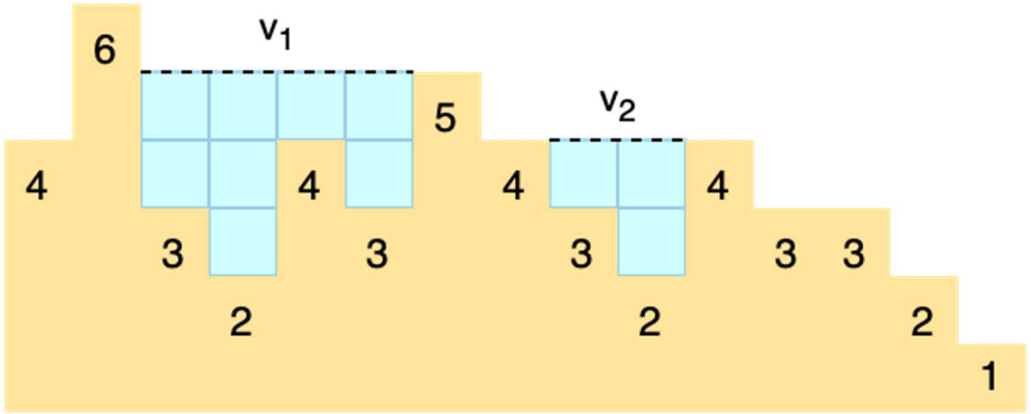
Output

For each pair of input, the program will provide one integer as output representing the number of the white adjacent tiles.

Sample input	Sample Output
1 9 2 BBBBBBBBB BBWWBBWBB BWWBBBBBB BWWWBWWBB BBBWWWBBB BBBBBBBBB BBBWWBBWB BBWBWBBBB BBBBBBBBB 3 2 7 5	12 4

PROBLEM 5: 2D Water Reservoir

Mr. Rhombus is the mayor of 2D World. He is planning on building a water storage reservoir in the countryside. Mr. Rhombus does not want to spend money digging a reservoir, so he would like to use the natural landscape to his advantage. He surveyed the countryside and recorded the landscape data in a quantized fashion, where each 1-unit length of land is assigned a height value ***h*** (denoted by number written on each bar in diagram):



Mr. Rhombus then proceeded to define "valleys". A valley is formed whenever two land units enclose an area such that the height of all the other land units in between those land units is shorter than them. In the diagram, two possible valleys are denoted as $\mathbf{V_1}$ and $\mathbf{V_2}$. Each valley has some capacity of storing water which is defined as the free area in the valley, such that height of the free area does not exceed the height of any of the enclosing land units of the valley. In the diagram, the capacity of $\mathbf{V_1}$ and $\mathbf{V_2}$ is 8 and 3 units respectively.

Mr. Rhombus came to you with the survey results and wants you to find him the perfect valley to be used as a water reservoir. The perfect valley, according to him, is a valley such that the capacity of storing the water in that valley is maximum, out of all the other valleys.

Input

First line of the test case contains one integer ***L***, which is the length of the countryside. Second line contains ***L*** integers, where each integer $\mathbf{H_i}$ denotes the height of the i^{th} land unit.

Output

Print a single integer denoting the maximum capacity out of all the valleys.

Limits

$1 \leq \mathbf{L} \leq 10^7$

$1 \leq \mathbf{H} \leq 8000$

Sample input	Sample Output
15 4 6 3 2 4 3 5 4 3 2 4 3 3 2 1	8

PROBLEM 6: Spray in Agriculture

Time limit: 3 seconds

A farmer in Neverhood has a large farm. He used to spray insecticides and pesticides manually. That used to cost him months to cover all his farm. Then he came across spraying drones that reduces the task to few hours only such that it sprays 1 liter of liquid per acre in 1 hour. The challenge with the drone technology was to refill the tank and battery at regular intervals. For that the drone has to fly back to origin, refill and recharge and then start the mission again. The farmer soon realized that as the drone started spraying further in the field, it has to cover already sprayed field and that costs lots of battery power to come to the base and return back to the location where it stopped spraying.

Although the farmer has invested a lot on purchasing the drone but seeing the benefits, he is ready to invest more in developing the infrastructure for more efficient spraying. At different locations in his field, he installed charging units and spray refill points, so that the drone does not have to return back to starting location for charging and refilling. Limited with resources, none of the locations have the same setup. At some places, charging capabilities are better and at some places spray refill is better. To compensate the differences, the farmer is asking you to develop an autonomous drone that can make minimum number of stops for recharging and refueling and complete spraying the whole field.

Input

The first line of the input consists of t, $(1 \leq t \leq 25)$ representing the total number of test cases given. The first line of the test case contains area to spray as n, $(1 \leq n \leq 10^9)$, starting fuel/spray as f, $(1 \leq f \leq 10^9)$, number of stations as s $(0 \leq s \leq 900)$ separated by a space. The next s lines contain site information with d $(1 \leq d \leq 10^9)$ as distance from origin and r $(1 \leq r \leq 10^9)$ as charge capability of the site.

Output

Output consists of t lines, each line contains exactly one number, representing the minimum number of refueling stops. If no solution exists, print -1.

Sample input	Sample Output
5 3 3 1 2 3 120 10 4 10 60 20 25 30 30 60 40 15 3 4 2 5 3 1 6 3 12 6 570 140 4 140 200 160 130 310 200 330 250 1360 380 5 310 160 380 620 700 89 850 190 990 360	0 3 4 2 2

PROBLEM 7: The art of war

Time limit: 3 seconds

You are the commander of an army of Fire Nation and you have many opponents to fight. All of your opponents have some power **P** gained by their victories. Some of your opponents have alliances with each other, i.e., they assist each other. By the recent information received from your spies all of your opponents are allied by all other opponents with power (**P - 1**) or power (**P + 1**). One opponent can have none or many allies.

If you defeat an opponent, you get all their power. But as you are not very strong yet, you do not want to fight them all. You are looking to use some war tactics. Your counselor has advised you to call a truce with the allies of the opponent that you are going to wage a war on, by forcing them to break the alliance with that opponent as a part of a peace deal. Be aware after calling the truce with a party you can't get their power. Also, many opponents can have the same power because they won the same number of wars.

Your obvious goal is to defeat as many opponents as possible and gather as much power as you can. You also must honor the advice of your counselor.

Input

- The first line contains the integer T, representing the number of total test cases.
- Each test case begins with an integer N that shows how many opponents you have
- Then a line follows containing N integers. $P_1, P_2, \ldots P_N$ in which P_i is the power of the i^{th} opponent.

Output

For each test case output as follows:

- Print a single integer — the maximum power you can achieve, assuming you have no power in the start.

Constraints

- $1 \leq T \leq 1000$
- $1 \leq N \leq 10^5$
- $1 \leq P_i \leq 10^5$

Explanation

Consider the third case (in bold) below. First, we need to select any opponent with power equal to 2. After that, all allies (with Powers 1 and 3) will come under a truce with us. Remaining opponents will look like this [2, 2, 2, 2]. Then we do 4 wars, in each war we select any opponent with power 2. In total, we get power equal to 10.

Sample input	Sample Output
3 2 1 2 3 1 2 3 **9** **1 2 1 3 2 2 2 2 3**	2 4 10