ICPC Masters Mexico 2020 - 2021

January 30th, 2021

Problems book

General Information

This problem set contains 13 problems; pages are numbered from 1 to 14, Without considering this page. Please, verify your book is complete.

A) Program name
1) Solutions written in C/C++ and Python, the filename of the source code is not significant, can be any name.
2) Solutions written in Java, filename should be: *problem_code*`.java` where *problem_code* is the uppercase letter that identifies the problem. Remember in Java the main class name and the filename must be the same.

B) Input
1) The input must be read from *standard input*.
2) The input is described using a number of lines that depends on the problem. No extra data appear in the input.
3) When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input.
4) Every line, including the last one, ends with an end-of-line mark.
5) The end of the input matches the end of file.

C) Output
1) The output must be written to *standard output*.
2) When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output.
3) Every line, including the last one, must end with an end-of-line.

Problem A

Adventure on the space

Professor JJ has developed the most powerful space travel machine, this machine has the ability it can go from planet A to planet B in just a few miliseconds if B is in the spatial range from A. The spatial range in a planet is given by an integer D, and it is said B is in the spatial range of A if the distance between the planets is at most D.

As some of you are aware JJ prefers always to go in the most secure way, this is why he is developing what he calls the most secure grid of planets. The most secure grid of planets is created selecting pairs of planets where you could travel between those planets using JJ's space travel machine and also it follows these restrictions:

- All planets in the universe should be in the grid of planets.
- For any pair of planets A and B there is only one way to reach B from A using JJ's space travel machine

You always argue with JJ, and now you two are arguing that he should not call his grid "the most secure grid of planets" as there can be more than one way to create it or there may not be a way to create the grid at all. JJ will not believe you until you get some evidence. This is why he just challenged you (as always) given the number N of planets in the universe, the value D that defines the spatial range of a planet and the coordinates where the planets are in the universe, can you determine how many different "secure grids" JJ can create? A "secure grid" A differs from other "secure grid" B if there is at least one pair of planets in the grid A that is not in the grid B.

Input

The input consist of several test cases. Each test case begins with a line containing two numbers N and D ($2 \leq N \leq 10$, $1 \leq D \leq 10^3$). The next N lines contains the values x_i, y_i ($0 \leq x_i, y_i \leq 10^3$) separated by a space the coordinates of each planet. The end of the test cases is given by a case where N and D equals 0.

Output

For each test case print in one line the number of different "secure grids" JJ can create. As this number can be very large please print it modulo $10^9 + 7$

Input example 1	**Output example 1**
3 5 1 1 2 2 1 2 3 1 1 1 2 2 1 2 3 1 1 1 2 2 3 3 0 0	3 1 0

Problem B

Buggy Text Processor

A string is traditionally a sequence of characters, either as a literal constant or as some kind of variable. In order to avoid data corruption, some text processor software will ignore any invalid characters from strings that are typed from a human interface device before storing the string into a database.

You found a bug on a popular open source text processor, this processor is intended to only allow letters as valid characters but it is also allowing dashes ('-') as valid input. You are working on a fix for this bug, and in order to test your fix your task is to count the number of characters that will be stored in the database, after typing N strings to the system.

A valir letter in an input string is a character representing any lowercase or uppercase letter from the english alphabet.

Input

The first line contains a single integer N ($1 \leq N \leq 100$), each of the next N lines contains a string with at most 100 characters. The strings in the input contain only dashes, uppercase and lowercase letters from the english alphabet.

Output

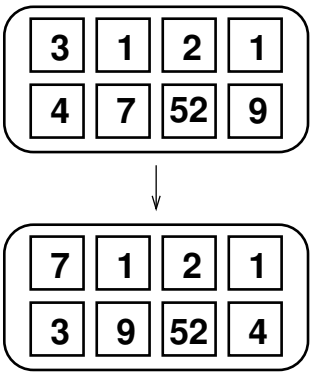
Print a line with a single integer, the number of characters that should be stored in the database.

Input example 1	**Output example 1**
2 ---- a-B	2

Input example 2	**Output example 2**
3 --- -- -	0

Problem C

Containers



The CBS–Container Balancing System needs to be updated so as to work with a new class of ships, the "two by four", which are ships that can carry eight large containers disposed in two lines and four columns, exactly as shown in the figure. These ships have a fixed crane that can perform a single type of movement: picking up two adjacent containers, in a row or column, and exchange their position. To speed up the loading process in the harbor, the eight containers are placed in any of the eight positions, defining an initial configuration. When ship leaves the harbor, the crane needs to move the containers so they end in a predefined final configuration.

The problem is that the cost for the crane to perform one movement is equal to the sum of the weights of the two adjacent containers whose positions are being exchanged. Given the weights of the containers in each position of both the initial and final configurations, the CBS has to compute the minimum possible total cost for a sequence of movements that leads from the initial to the final configuration.

Input

The input consists of four lines containing, each one, four integers between 1 and 1000, inclusive. The first two lines define the weights of the initial configuration and the last two lines, the weigths in the final configuration. There is always a solution, as the containers in the initial and final configurations are the same, possibly in different positions.

Output

Output a line containing an integer, representing the minimum possible total cost for a sequence of movements that leads from the initial to the final configuration.

Examples

Input example 1	**Output example 1**
3 1 2 1 4 7 52 9 7 1 2 1 3 9 52 4	81

Input example 2	**Output example 2**
1 2 3 4 5 10 7 8 1 2 3 4 5 8 7 10	50

Input example 3	**Output example 3**
34 5 6 998 4 17 77 84 34 5 6 998 4 17 77 84	0

Problem D

DIOS primes

In number theory, a left-truncatable prime is a primer number which, in a given base, contains no 0, and if the leading ("left") digit is successively removed, then all resulting numbers are prime. For example, 9137 is a left-truncatable prime, since $9137, 137, 37, 7$ are all prime. For this problem, decimal representation is assumed.

A Dios prime is a left-truncatable prime such that the number obtained by reversing its digits is also left-truncatable. The number 3467 is a Dios prime, since 3467 is left-truncatable and 7643 is also left-truncatable. Given two numbers a and b, your task is to find the number of Dios primes in the range $[a, b]$.

Input

The first input line contains a number T ($1 \leq T \leq 10^5$), the number of test cases. Each of the following T lines contains two numbers a and b ($1 \leq a \leq b \leq 10^9$) separated by a space.

Output

For each test case you must print the number of Dios primes in the range $[a, b]$.

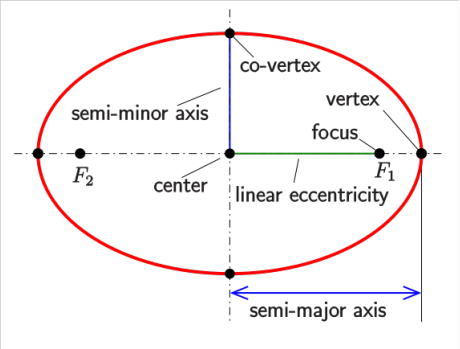
Input example 1	**Output example 1**
4 1 10 11 100 3467 3467 7643 7643	4 2 1 1

Side Remark. In order to dial *DIOS in a telephone, one has to dial *3467. That's how the Dios primes got their name.

Problem E

Ellipses

An ellipse is a plane curve surrounding two focal points F_1 and F_2, such that for all points on the curve, the sum of the two distances to the focal points is a constant D.



Given the coordinates of the two focal points (F_{1_x}, F_{1_y}), (F_{2_x}, F_{2_y}), and the constant D, your task is to find the number of points with integer coordinates that are strictly inside the ellipse defined by these values.

Input

The first line contains a single integer T $(1 \leq T \leq 100)$, the number of test cases. Each of the next T lines contains 5 integer numbers separated by a space, representing the values F_{1_x}, F_{1_y}, F_{2_x}, F_{2_y} $(-100 \leq F_{1_x}, F_{1_y}, F_{2_x}, F_{2_y} \leq 100)$, and D $(1 \leq D \leq 200)$, respectively.

Output

For each test case in the input print a single line with a single integer number, the number of points with integer coordinates that are strictly inside the ellipse defined in the test case.

Input example 1	**Output example 1**
2 0 0 0 0 4 -3 0 3 0 10	9 59

Problem F

Finding the Train

In a not so far future, trains are created totally using magnets. There are magnetic blocks that can be attached side by side regardless it's polarity, however they expose a polarity in the upper side of the block and on the lower side of the block, in such way that it allows an easier storage as if they are well designed you could break the train in two parts and stack one over the other.

As you can imagine by now a block can be stacked above another block if their polarity is not the same, but as these boxes are too heavy once the boxes are stacked in this way they can not be detached, however if a box is put above another box with the same polarity then the box will levitate over the other without moving.

Years, maybe centuries of work and effort have been put to create a machine that creates trains that can minimize the storage needed for them, a train is optimal for storage if it can be broken at the middle point so you have two equally sized parts that can be stacked one over the other in such way that the part you put above levitates, so that the length required to store the train is half the size of the train. As the half of the train that will be put above may be too heavy, the machine flips the two halves in such way that the last block of the first half is below the first block of the second half and the first block of the first half is below the last block of the second half.

Today the machine is not working properly and most of the trains turn out to not be optimal. You could easily disassemble all blocks from a train and return them to the machine to create a new train, but you risk to have another train that is not optimal for storage. So you take some initiative and decided to find from each train that the machine is creating what is the largest optimal storage train that you can get after removing an amount of blocks (maybe 0) from each of the sides of the train.

Input

The first line of input contains a number T ($1 \leq T \leq 10$), the number of trains to test. The following T lines contain a string S ($1 \leq |S| \leq 10^6$) representing the specification of a train, the string contains only the symbols '+' and '-' and represents the polarity of each of the blocks that create the train.

Output

For each train in the input your program should print a line with a single integer, the length of the largest optimal storage train that can be found on S removing blocks only from the sides of the train.

Input example 1	**Output example 1**
3 ++--++++--++- +-++-+ -++	12 6 2

Problem G

Game

The exponential points game is a two person board game. The board consists of N tiles numbered from 1 to N arranged over a straight line. Before the game begins, both players agree in a set of K tiles to be marked so that none of the players will take any of those tiles in the game.
The two players will alternate turns. On each turn, the player marks 1 or more unmarked tiles, but if the player marks more than one tile, all tiles marked in the turn should share a side (tile i is adjacent to tile $i-1$ and to tile $i+1$). In turn k, if the player marked m_k tiles then he scores $m_k \times 2^{k-1}$ points. The game continues until all the boxes are marked. As you can guess, the player with higher score wins the game.
Santiago is a fan of this game so he has challenged you to play. Since he is more experienced in the game, he says he will take the first turn, and also, he won't take any points from that turn to give you a chance to win. Then in this game, the first turn, Santiago will get $m_1 \times 0$ points, in the second turn you get $m_2 \times 1$ points, the third turn Santiago gets $m_3 \times 2$ points, the fourth turn you get $m_4 \times 4$ points, etc.
What Santiago does not know is that you are an expert in the game as well, so, you suspect Santiago is not giving a chance to win taking the first turn, instead, he is ensuring you will lose. You know both of you will play the game optimally, can you determine the minimum number of tiles Santiago has to mark in the first turn so that you don't have a chance to win?

Input

The first line of the input contains two numbers separated by a space N ($1 \leq N \leq 1000$) and K ($0 \leq K \leq N$), representing the number of tiles in the board and the number of tiles marked before the game starts, respectively. The following K lines contain each a number between 1 and N, inclusive representing the tiles that are marked before the game starts. No tile is marked twice in the input.

Output

Your program must output a single line, containing an integer indicating the minimum number of tiles to take in the first turn so that Santiago wins the game. If no such number exists, print -1., representing

Input example 1	**Output example 1**
4 0	2

Input example 2	**Output example 2**
5 3 1 2 3	-1

Input example 3	**Output example 3**
8 3 1 5 7	1

Problem H

HTML From Shortcuts

One kind of software that is becoming very popular is software for blog writting. A typical blog writting software will parse text into Hypertext Markup Language (HTML) that is suitable to display content in widely used web browsers. While you can write HTML, most content authors do not have expertise using it, or do not feel comfortable writing using HTML tags. To make their lives easier, you decided to offer a software with a simpler syntax, authors can write their content and use "shortcuts" to achieve some markup textual effects when displaying it. To start, you decided to implement "shortcuts" for two of the most used text decorations in a blog: bold text, and italic.

To write bold text, the author can write the asterisk ('*') shortcut, for example, if the author types: "*This is bold*", the software will render the HTML code "This is bold".

For italic, the author can write the underscore ('_') shortcut, for example, if the author types: "_This is italic_", the software will render the HTML code "This is italic".

To make the authors job easier, nesting of shortcuts is not allowed, this is no text enclosed in a shortcut will have another shortcut.

Your job is to write the code that will take a document written with shortcuts and translate it into proper HTML.

Input

The first line contains a single integer T ($1 \leq T \leq 100$), the number of test cases. Each of the next T lines describe a document written in shortcut. A document written in shortcut is a string with at most 100 characters, containing only characters that are the letters from the english alphabet: 'a' to 'z' and 'A' to 'Z', the underscore '_', the asterisk '*', the space character, and the punctuation symbols: ',', ';', '.', '!', '?', '-', '(', and ')'. It is guaranteed that if there will be an even number of underscores and asterisks.

Output

For each test case in the input print a single line containing the HTML code for the shortcut code given in the input.

Input example 1

```
2
_This is italic_*This is bold*
*(Is,;*this.)*a!*_question?_*-*__
```

Output example 1

```
<i>This is italic</i><b>This is bold</b>
<b>(Is,;</b>this.)<b>a!</b><i>question?</i><b>-</b><i></i>
```

Problem I

Interesting DNA Sequences

A DNA sequence is represented as a string where each character represents one of the four different bases in the molecular composition:

- A: Adenine
- C: Cytosine,
- G: Guanine,
- T: Thyamine

An example of DNA sequence then would be the string "AACGGT".

A development group in your work is interested in very specific DNA sequences, they know exactly the amount of times each base appears in the molecular composition, and also know that these sequences do not have two of the same base in adjacent positions, for example, if the sequence contains only 2 occurrences of adenine, and 2 occurrences of cytosine, then, "ACAC" is a sequence they are interested in, while "AACC", and "ACCA" are not. What they do not know is how many different DNA sequences may be of their interest. Can you help the group counting them?

Input

The first line of input contains a single integer T ($1 \leq T \leq 100$), representing the number of test cases. Each of the next T lines contains four integers separated by a space, representing respectively the amount of times each of the bases A, C, G, T ($0 \leq A, C, G, T \leq 10$), appear in the DNA sequence of interest for the development group.

Output

For each test case in the input output a line containing a single number, the number of different DNA sequences that may be of interest to the development group.

Input example 1	**Output example 1**
4 0 0 0 0 1 1 0 0 1 1 1 5 1 1 1 1	1 2 0 24

Problem J

John's New Product

John's transportation Company is very close to release their new product, an amazing drone able to go to market and buy groceries for their owners. John's company has two product candidates for release: the candidate A with a sunstained battery life, and candidate B with enough power to make groceries faster. To decide which one will get to market, they will perform A/B testing of the product in N cities of the country, giving the product for freee to selected testers on each city.

A/B testing is a way to compare two versions of a single variable, typically by testing a subject's response to variant A against variant B, and determining which of the two variants is more effective.

The way the test will be performed is giving either candidate A or candidate B of the product to the testers of each of the N cities, in such way, all testers on each city will test only product candidate A or product candidate B. After one month of testing, the test will end and they will run a survey on each tester who got the product to perform some analysis and determine which candidate turned out to be more effective for release. John believes the best way to distribute the candidates should follow two rules:

- If there is a road connecting city u and city v, then candidate offered on city u and city v should not be the same.
- The difference between the number of people who tested candidate A and the number of people who tested candidate B should be the minimum possible.

John needs your help to determine which product candidate should be offered on each city. Given the information of cities connected by roads, and the number of testers identified on each city, find the minimum possible difference of number of testers between product candidates.

Input

The first line of input contains two integer numbers N and R ($1 \leq N \leq 100$, $0 \leq R \leq \frac{(N)(N-1)}{2}$), representing the number of cities and the number of roads in the country. The second line contains N integer number separated by space, where the i-th number represents the number of testers on city i, the number of testers in each city will be between 0 and 100. Each of the next R lines contains two integer numbers u and v separated by a space, representing there is a road connecting cities u and v, if a road connects cities u and v, it also connects cities v and u. You can assume it is always possible to assign the product candidates on the cities following John's rules.

Output

Output a line containing one integer number, the minimum possible difference between the number of testers for product candidate A, and the number of testers for product candidate B.

Input example 1	**Output example 1**
3 0 5 5 12	2

Input example 2	**Output example 2**
4 2 2 4 2 4 1 2 3 4	0

Input example 3	**Output example 3**
3 2 5 1 5 1 2 2 3	9

Problem K

K Scores

Cherries harvesting season has finished and The International Cherries Packaging Company (ICPC) have packaged the collected cherries into N packages. Before labeling the packages for distribution, they perform some tests on them, one of the tests will score each package with an integer number between 1 and 10^6, this score represents the quality of the cherries in the package, the smaller the score, the better the package. After all the packages are scored, they are placed in a line and identified with the numbers from 1 to N in increasing order from left to right, this is, the leftbox package in the line has the index 1, the one to its right has index 2, until the rightmost package with index N.

ICPC is interested in the automation of the second test that is performed to packages, while the first test is performed individually to each package, the second test is performed to groups of contiguous packages in the line. They select two indexes L and R ($L \leq R$), and a random value K ($1 \leq K \leq R - L + 1$), the test should determine the score S_k of the range which is the K-th smallest score from the packages with indexes between L and R inclusive, they are also interested in the sum of the scores of all packages in the range with a score greater to S_k. Your task is to write a program that finds these two values for several tests that will be run on a given set of packages.

Input

The first line contains two numbers N ($1 \leq N \leq 10^6$) and T ($1 \leq T \leq 5 * 10^5$), representing the number of boxes and the number of tests to perform. The next line contains N integer numbers separated by a space, where the i-th number represents the score of the package with index i. Each of the next T lines contains three integer numbers separated by a space L, R, and K ($1 \leq L \leq R \leq N$, $1 \leq K \leq R - L + 1$)

Output

For each test in the input print a line with two numbers separated by a space, the score S_k of the given range, and the sum of scores of all packages in the range with a score greater to S_k .

Input example 1	**Output example 1**
5 3 3 1 2 5 3 1 1 1 1 5 3 2 4 2	3 0 3 5 2 5

Problem L

Lost Card Game

Lost card game is a game that is played by a single player, it consits of a stack of N cards, each card has 3 different uppercase letters drawn on it. Given two of these cards, their similarity is defined as the number of letters they have in common, for example, the cards "ABC" and "CBD" have similarity 2 because the letters 'B' and 'C' appear in both cards, while the similarity for "ABC" and "DEF" is 0 because no letter is common in both cards.

You can remove a card from the stack if the similarity of the card above it and the card below it in the stack is at least 2. Once the card is removed from the stack, the card below it will be below the card that was above the removed card and viceversa, as you can see, the card at the top will never be removed since it does not have a card above it, in the same way, the card at the bottom will neved be removed since it does not have a card below it. The objective of the game is to remove the maximum number of cards you can remove from the stack. Write a program that given the cards in the stack finds the maximum number of cards you can remove from it following the rules above.

Input

The first line of input contains an integer T ($1 \leq T \leq 100$), representing the number of test cases that follows. Each of the next T test cases contains of two lines, the first line contains a number N ($2 \leq N \leq 100$), representing the number of cards in the stack. The second line contains N strings, each string with exactly three differente uppercase letters, representing the i-th card in the stack, the first card is the one at the top of the stack, the last one is the one at the bottom.

Output

For each test case in the input output a line with a single integer, the maximum number of cards that can be removed from the hand.

Input example 1	**Output example 1**
3 2 ABC CBA 5 ABC CDE ABC CDE ABC 3 ABC BCD CDB	0 3 1

Problem M

Managing Race Results

Horse racing is an equestrian performance sport, typically involving two or more horses ridden by jockeys (or sometimes driven without riders) over a set distance, for competition. It is one of the most ancient of all sports, as its basic premise – to identify which of two or more horses is the fastest over a set course or distance – has been unchanged since at least classical antiquity.

You have been following the results of several horse races, and identified that every saturday the same N horses identified with the numbers 1 to N will race. People is usually interested in identifying which horse will win a race, since they can get a lot of money from gambling, you are interested not only on knowing what will be the first horse to finish the race, but, on what will be the position each hore will finish. In order to achieve this, you have been following the performance of the horses, and have identified a list of M pairs of horses u, v, such that the horse u will always finish the race before the horse v. Given the number of horses in the race, and the pairs of horses you have identified, find how many final results are possible when the race ends assuming the observations in your list will not change.

Input

The first line contains an integer T ($1 \leq T \leq 100$), representing the number of test cases. T test cases follows, each test case starts with a line containing two integer numbers N and M, representing the number of horses on the race, and the number of pairs in your list ($1 \leq N \leq 30$, $0 \leq M \leq 15$), the next M lines of the case contains two integer numbers separated by a space u, and v ($1 \leq u, v \leq N$, and $u \neq v$), representing the horse u always finishes before horse v.

Output

For each test in the input print a single line with an integer, representing the number of different possible results for the race that hold the observations from your list. As this number can be very big print it modulo $10^9 + 7$.

Input example 1	**Output example 1**
3 3 1 2 3 4 2 1 2 1 3 5 3 2 3 3 4 4 2	3 8 0

Side Remark. In the last test case the answer is 0 because there is no possible result that satisfies all observations from the list.