Gran Premio de México 2020 - Primera fecha

October 31st, 2020

Problems book

General Information

This problem set contains 14 problems; pages are numbered from 1 to 21, 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.
3) Solutions written in Kotlin, filename should be: *problem_code*`.kt` where *problem_code* is the uppercase letter that identifies the problem. Remember in Kotlin 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

Advanced Recommendation System

You are working on a new recommender system for an important marketplace company which sells N different items with integer identifiers from 1 to N.

What the recommender system does is predict the preference a user would give to an item based on a list of other items he already bought. The system is used to predict what item j a user would most likely react after having reacted to a product i. If the recommender system finds evidence that the user reacts to a product j after reacting to the product i , then the preference of j over i increases by 1 which means the system will more likely recommend j to users that have previously reacted to i.

The marketplace found this recommender system is working quite well, however, they identified that sometimes given the product i the system recommends a product j which is not the product with more preference over i, this is, there is another product k that has been reacted more times after reacting to product i than product j. They want you to make a change so that the product that is recommended for product i is the one with the most preference among the products users have reacted after reacting to i. They are also interested to answer if there are more than one of such items, and to find among all pairs of items (i,j) what pair is the one where the preference j over i is maximum in the system, in case such pair exists. Note that the preference for j over i is not necessarily the same as preference for i over j.

Input

The first line of input contains two integer numbers separated by a space, N and Q ($1 \leq N, Q \leq 2 \times 10^5$)representing the number of items in the marketplace and the number of queries that will be performed to test your system. Each of the next Q lines describe a query to your system, being one of the following:

- R i j: The system registered evidence that someone reacted to the product j after i.
- Q i: Answer the query, what is the product j that should be recommended.
- B: Answer the query, what is the pair i, j where the preference for j over i is the maximum.

Output

For each query 'Q' answer a line with the id j of the product that should be recommended, if there are multiple products that could be recommended print the string "Multiple", if the system does not have enough information to answer print "No info". For each query 'B' answer a line with two numbers separated by a space representing the values for i and j where the preference for j over i is the maximum, if there are more than one such pairs print "Multiple". The queries should be answered in the same order they appear in the input

Input example 1	**Output example 1**

`5 13`
`R 1 2`
`Q 1`
`Q 4`
`R 2 3`
`R 3 2`
`R 2 3`
`Q 2`
`B`
`R 1 4`
`R 1 5`
`R 1 4`
`R 1 2`
`Q 1`
	`2`
	`No info`
	`3`
	`2 3`
	`Multiple`

Problem B
Bus Line

In Jaime's city people who uses public transportation waits in line on the bus station to get on a bus. Once the bus arrives, people get on the bus in the order they were on the line, they always get on the bus using the front door, they pay, and then walk to the backdoor of the bus until they reach the backdoor or a person, so the person that gets on the bus first is the closest to the backdoor of the bus (until that person gets off). Unlike most of public transportation systems, buses at Jaime's city do not have seats, this is why people stands side by side on a line waiting to reach their station to get off.

Bus rules dictate that everyone should get off using the backdoor, this is troublesome for some people, but very annoying for Jaime. Jaime is usually the last one getting on the bus, and when he has to get off, to reach the backdoor and step off, he needs to swap places with each person that took the bus before him. More generally if Jaime needs to get off the bus before someone that got on the bus before him, Jaime will need to swap places with that person to get off. Jaime filled a complaint arguing that it is better to use the nearest door to get off the bus, that way the number of swaps to get off the bus can be minimized, in this case Jaime would not swap places with anyone, if he was the last person getting on the bus, at the time he needs to get off he could just get off from the front door as he is the closest one to it.

The bus station answered Jaime's complaint requesting him evidence that this method would be better. As showing this is better for him is not enough evidence, Jaime will find evidence that in general this is better to anyone using the bus. Jaime will ask each person in the line at what bus station they need to get off so that he can calculate the total number of swaps needed before all people gets off the bus if only the backdoor is allowed, and the total number of swaps needed before all people gets off the bus if it is allowed to get off using either the front or the backdoor.

Jaime has the list, but, is having a hard time finding these numbers, can you help him with that task? You can assume people will not get on the bus after leaving Jaime's station which is station 0, and also the bus arrives to the stations in order, this is it first gets to station 1, then station 2, etc...

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^5$), representing the number of people that will get on the bus in Jaime's stop. Each of the next N lines contains a single integer number b_i ($1 \leq b_i \leq 10^5$), representing the station at wich the $i-th$ person in line needs to get off the bus.

Output

Output a single line with two numbers separated by a space, representing the total number of swaps needed before all people gets off the bus if only the backdoor is allowed, and the total number of swaps needed before all people gets off the bus if it is allowed to get off using either the front or the backdoor, repectively.

Input example 1	**Output example 1**
6 2 5 6 3 8 4	5 2

Problem C
Continuous Replacement Algorithm

Alice and Bob are planning to have a secret way to communicate within each other, they know it is not a good idea to create their own cryptographic algorithm, but they insist they need to do something new as they do not want anyone to understand what they are saying in their messages if any gets intercepted.

They have come with a simple idea: Alice created a list of pair of words that can be substituted in the message, this is, if the list contains words u and v, then the word u can be replaced by v, and the word v can be replaced by u in the message. What they will do to communicate is write the message they want to communicate and start replacing the words following the substitution list. If after replacing all the words, the resulting message has words that can be replaced they will repeat the process, they do this making sure that after replacing the words they always get a smaller message. The process will stop once the message can not be smaller replacing any of the words.

Alice and Bob found this is a very difficult process to be done manually, so they have come to you asking for help, given the list of words for substitution and a message to communicate output the message after applying the substitution process.

Input

The first line of input contains a single integer number N the number of pairs of words in the substitution list ($1 \leq N \leq 10^5$), each of the next N lines contains two words separated by a space u and v ($1 \leq |u|, |v| \leq 15$), representing that the word u in the message can be substituted with the word v or vice versa. The next and last line of input contains the message m to be communicated ($1 \leq |m| \leq 2 \times 10^5$).

Output

Output a line containing the message after applying Alice and Bob substitution process. In case there are multiple possible messages print the one with the smallest amount of characters, if there are multiple messages with the smallest amount of characters, print the lexicographically smallest.

Input example 1	Output example 1
8 sea see see c you u and an n an are r ok k k z i sea you and you are ok	i c u n u r k

Problem D
Detailed Sorting Machine

In computer science, a sorting algorithm is an algorithm that puts elements of a list in a certain order. The most frequently used orders are numerical order and lexicographical order. Efficient sorting is important for optimizing the efficiency of other algorithms (such as search and merge algorithms) that require input data to be in sorted lists. Sorting is also often useful for canonicalizing data and for producing human-readable output. More formally, the output of any sorting algorithm must satisfy two conditions:

- The output is in nondecreasing order (each element is no smaller than the previous element according to the desired total order);
- The output is a permutation (a reordering, yet retaining all of the original elements) of the input.

Your school is running the Sorting Algorithms Week (SAW), in this week some clever engineers create clever machines to demonstrate how sorting algorithms work in a more physical way. This years winner created a machine that not only performed sorting but allowed people to interact with the machine to better understand the whole sorting process. As usual the winner will present the machine to the crowd.

The machine has a display where N boxes can be seen sitting on N positions numbered from 1 to N in a straight line from left to right, each box has a number B_i drawn and visible to the espectators of the show. The way people interacts with the machine is through N buttons that are on the outside of the machine, the buttons are numbered from 1 to N and are aligned to the positions where the boxes sit so that the button with the number 1 is aligned with the first (leftmost) position, the second button is aligned with the second position, ..., and the $N-th$ button is aligned with the $N-th$ (rightmost) position. When a person pushes the $i-th$ button, the box at the $i-th$ position in the machine is lifted by a crane, all other boxes are moved one place to the left, and the box that was lifted is put down on the $N-th$ position.

The engineer who created the machine claims there are several ways to push the buttons so that the numbers drawn on the boxes are sorted in ascending order. Given the number of boxes and the values B_i each box has visible before any button has been pressed. Can you find the minimum number of times the buttons should be pressed to sort the values drawn in the boxes in ascending order?

Input

The first line of input contains a single integer number N ($1 \leq N \leq 10^5$), representing the number of boxes in the machine. The second and last line contains N integer numbers separated by a space, where the $i-th$ number represents the value B_i ($1 \leq B_i \leq 10^6$) drawn in the box sitting on the $i-th$ position of the machine.

Output

Print a single line with a single integer, representing the minimum number of times the buttons should be pressed to sort the values drawn in the boxes in ascending order.

Input example 1	Output example 1
5 1 3 5 7 9	0

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

Problem E
Enthusiastic Mathematics Challenge

Bob, a mathematics enthusiast, studying everyday number theory and other topics always finds a way to challenge Alice, a computer science enthusiast. You always see with amusement how Bob challenges Alice, and how she always finds a way to build a computer program to solve Bob's challenge, however, this time, Bob found a challenge so hard that Alice was unable to solve, even with a program.

Bob wrote a string S containing only digits, he says there are 3 prime numbers hidden in S, that those prime numbers can be found as three disjoint sets from S, and that all the digits of S should be used, Bob wants Alice to find what the product of those 3 prime numbers is.

You have seen Alice trying to find the answer to Bob's challenge and decided to help her, after a lot of time invested you found that there is no way to find the 3 prime numbers following Bob's rules. This is why you decided to help Alice to write a program that finds the 3 prime numbers if they exist.

Input

The input contains a single line with a string S $(1 \leq |S| \leq 8)$ containing only digits.

Output

Output a line containing a single integer, the product of the 3 prime numbers if they exist, if there are more than one possible solution print the smallest one. If S does not contain the 3 prime numbers output a line with the string "Bob lies"without the quotes.

Input example 1	Output example 1
`333`	`27`

Input example 2	Output example 2
`707070`	`Bob lies`

Input example 3	Output example 3
`157`	`Bob lies`

Input example 4	Output example 4
`01123`	`606`

Problem F
Feeding The Judges

As you know, there is an important programming contest today where several brilliant teams are competing. In addition to the contestants, a contest needs judges, who, in this case, will have to work hard all day long. At this moment, they all are in some remote building where they have been provided of enough food and a lot of things to do.

Like any other human being, judges dislike some food. For this reason, before their food was bought, judges were asked to specify which dishes, and how many of them (at least), would like to have; they wrote down in a list the number of pieces of every dish followed by a space and the name of the dish, but they are so busy that they may wrote words mixing uppercase and lowercase letters. You can be sure every judge requested at least one dish, and that all dishes are available in the restaurant.

On the other hand, the restaurant has available K different dishes, and for each dish they have a maximum amount of plates they can serve within the contest timeframe.

Your task is to find in how many different ways it is possible to place an order for the restaurant to serve exactly N dishes making sure every judge will have what they requested, and the quantity of every dish bought does not exceed the amount of plates the restaurant can serve for that dish.

Input

The first line of input contains a single integer number N ($1 \leq N \leq 2 * 10^5$), representing the amount of plates that should be ordered; then, you will receive several sets of lines (at most 100, at least 1), each set representing what a judge requested: the first line on each judge request contains a string representing the name of the judge (up to 10 characters), followed by at least one line that describe the judge request, each of these lines will have an integer number ($1 \leq d_i \leq 10^4$), representing how many plates of this dish the judge wants followed by a space and a string representing the name of the dish the judge wants (up to 20 characters). You will know the list of judges ended when you find the line with word "Restaurant", followed by at most 15 lines, describing the dishes available in the restaurant, each of these lines contains an integer number b_i ($1 \leq b_i \leq 10^6$) representing the maximum amount of plates the restaurant can serve for this dish, a space and the name of the dish wrote correctly in lowercase letters.

Output

Print a single line with an integer number representing the number of different ways it is possible to buy the judges food satisfying all the mentioned restrictions. As the answer could be very big, print it modulo 188888881.

Input example 1	Output example 1
12 juan 1 sALAd 3 pASta 2 Fish carlos 2 salAD laura 1 salad 1 PaStA Restaurant 5 salad 5 pasta 4 fish	4

Input example 2	Output example 2
18 juan 1 saLad 3 pASta 2 fiSh 1 dEsSeRt carlos 1 chiCKen 3 salad laura 1 SalaD 1 pasta 3 DessERT Restaurant 2 salad 5 pasta 4 fish 5 dessert 2 chicken	0

Problem G
Golf Score

Golf is a club-and-ball sport in which players use various clubs to hit balls into a series of holes on a course in as few strokes as possible.

Unlike most ball games, golf cannot and does not utilize a standardized playing area, and coping with the varied terrains encountered on different courses is a key part of the game. The game at the usual level is played on a course with an arranged progression of 18 holes.

A hole is classified by its par, which gives an indication of the number of strokes a skilled golfer may be expected to need to complete play of the hole.

The goal is to play as few strokes per round as possible. A golfer's number of strokes in a hole, course, or tournament is compared to its respective par score, and is then reported either as the number that the golfer was "under" or "over" par, or if it was "equal to par". Common scores for a hole also have specific terms:

Term	Description
Hole in one	Exactly one stroke
Condor	Four strokes under par
Albatross	Three strokes under par
Eagle	Two strokes under par
Birdie	One stroke under par
Par	Exactly par
Bogey	One stroke over par
Double bogey	Two strokes over par
Triple bogey	Three strokes over par

The most prestigious golf course in the city has hired you to build a golf course management system, one of the most important features in this system is to calculate the total number of strokes a player did in a round given the term that describes the player performance on each of the 18 holes in the course. Write a program that given the par for each hole and the term that describes the player performance on each hole return the total number of strokes the player performed in the round.

Input

The input contains exactly 19 lines. The first line of input contains 18 integer numbers separated by a space where the $i-th$ number represents the par for the $i-th$ hole in the course. Each of the following 18 lines contains a lowercase string describing the performance of the player in that hole (from the table above). It is guaranteed that the par for a hole does not exceeds 5.

Output

Output a line containing a single integer number representing the number of strokes the player performed in the round.

Input example 1	**Output example 1**

`1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 4 4`
`par`
`par`
`par`
`par`
`par`
`par`
`par`
`par`
`par`
`par`
`triple bogey`
`double bogey`
`bogey`
`par`
`birdie`
`eagle`
`albatross`
`hole in one`
	`45`

Problem H
Halloween

Halloween is here and Baker is planning to collect as many candies as possible! He has been studying his neighborhood, he knows that all the houses are lined up one next to each other and the amount of candies he can get from each house. This has been a very consistent number through the years.

Everyone in the neighborhood is concerned about Baker's health so they made an agreement: if Baker comes Trick-or-treating to their door and they saw him in either one of the adjacent houses, they will not give him any candy.

Given these conditions, Baker wants to know the maximum number of candies he can get on Halloween's night.

Input

The first line of input contains an integer T $(1 \leq T \leq 50)$ representing the number of test cases in the input. T test cases follow, each test case consists of two lines of input, the first one contains an integer N $(1 \leq N \leq 50)$, representing the number of houses in Baker's neighborhood, the second line of each test case contains N integer numbers, where the $i - th$ number in the line represents the amount of candies C_i $(0 \leq C_i \leq 100)$ Baker can get from the $i - th$ house in his neighborhood.

Output

For each test case, output a line containing a single integer, the maximum amount of candies Baker can get this Halloween.

Input example 1	**Output example 1**
3 3 1 2 3 8 9 21 33 31 21 84 86 53 2 83 34	4 189 83

Problem I
Is It Secure Enough?

As part of the authentication system your company is working on you should write a password validator. The main intention of this is advise users about how strong is the password they captured in a text box. The rules used to calculate the password strength are the following:

- At least one lower case
- At least one upper case
- No consecutive numbers together (digits), and at least one number, regardless the order
- At least one special char among the set '.', '#', '$', '%', '/', '(', '&', ')'
- At least 10 characters long

Your validator should show the user the password strength based on the following classification:

- Strong: Complies with all rules
- Good: Complies with 4 rules
- Weak: Complies with 3 rules
- Rejected: Does not comply with at least 3 rules

Before putting the validator in production you need to run some tests on it. Given a list of passwords provide the password strength classification for each of them.

Input

The first line contains a single integer N ($1 \leq N \leq 10^5$)representing the number of passwords to test. Each of the following N lines contains a string s representing a password to test ($1 \leq |s| \leq 100$). It is guaranteed the password does not contain whitespaces and all characters will be lowercase letters, uppercase letters, all letters will be from the english alphabet, digits, or any of the characters '.', '#', '$', '%', '/', '(', '&', ')'.

Output

For each of the N passwords in the input print a line, representing the password strength classification (Strong, Good, Weak, or Rejected), with the Following format *Assertion number #<case number >: <classification >*

Input example 1	Output example 1

`4`
`Password`
`myColl#3Pa.s.word`
`WeaK.`
`Good.1`
	`Assertion number #1: Rejected`
	`Assertion number #2: Strong`
	`Assertion number #3: Weak`
	`Assertion number #4: Good`

Problem J
Jaime's Room

Jaime has a very messy room with N objects, in the past, Jaime used to be very careful with his belongings that is why each object has a label that uniquely identifies it with the numbers from 1 to N. Jaime has been very busy training for the next programming contest, and his mother, seeing how busy he is, decided to clean the room with the only condition that it will be cleaned in the way she finds it best, using boxes and throwing some things to the thrash, to not be bothered with his training Jaime agreed to his mother's terms.

Jaime's mother has C boxes labeled with the numbers from 1 to C that she will use to clean Jaime's room, each day she decides to move some objects to a box, to make her work easier (and harder) she takes the box with label k and picks two numbers i and j then she proceeds to move all objects identified with the numbers i, $i+1$, ... j to the box, regardless if they are already in another box. Since Jaime has a lot of items, sometimes Jaime's mother decides to throw a box to the thrash with all the objects inside of it, once the box have been thrown neither Jaime nor his mother will be able to find the object.

With all the changes Jaime's mother is doing in his room, Jaime is unable to find objects. He decided to create a program to help him find them using the information his mother gives him regarding what objects have been moved to what box and what boxes have been thrown to the thrash. Help Jaime write this program to answer what box has a given item and if given two items they are on the same place (same box, room, or thrash).

Input

The first line of input contains two integer numbers N $(1 \leq N \leq 10^5)$ and C $(1 \leq C \leq 10^7)$ separated by a space, representing the number of objects in Jaime's room, and the number of boxes his mother has available. The next line contains a single integer Q $(1 \leq Q \leq 10^5)$ representing the number of actions to perform in the program. Each of the next Q lines describe an action to be performed in the program, the actions can be one of the following:

- m i j k : Move all the objects with labels i to j (inclusive) to box k
- b k : Throw box k to the thrash with all its contents
- d i : Answer the label of the box that contains the item with index i
- s i j: Answer if the object with index i and object with index j are in the same place (same box, room, or thrash)

It is guaranteed that for any action: $1 \leq i, j \leq N$ and $1 \leq k \leq C$.

Output

For each action that starts with 'd' print a single line with an integer representing the box that contains the object, if the object is in the thrash print the string "para que quieres saber eso", if the object is in the room but not inside a box print the string "si lo encuentro que te hago". For each action that starts with 's' print a single line containing the string "si" if the two objects are in the same place, or the string "no" otherwise. Answer to the actions should be in the same order as the actions appear in the input

<table>
<tr><th>Input example 1</th><th>Output example 1</th></tr>
<tr><td><pre>10 5
5
m 1 4 2
m 3 5 1
s 4 5
b 2
d 1</pre></td><td><pre>si
para que quieres saber eso</pre></td></tr>
</table>

Problem K

King's Dilemma

Quadradonia's king is willing to repair some of the roads that connect his kingdom. The kingdom has N cities and M roads, each road connects two cities u, and v, in such way that you can travel from u to v and from v to u. As Quadradonia's has invested in several infrastructure projects during the last year it is not possible to repair more than K roads, so the king decided to repair exactly that number of roads, no more, no less.

There are some cities that have roads with more issues than others, that is why the king decided to classify the cities by how much they need the roads to be repaired and use that information to decide what roads to repair. A needness index has been set to the cities numbering them from 1 to N, so that the city with the number 1 has the most need for the roads to be reparied, and the city with the number N has the less need. Using these indexes the needness for a road to be repaired can be stated as follow: if a road R_1 connects cities with indexes u and v where $u < v$, and another road R_2 connects cities with indexes x, and y where $x < y$, then R_1 has more need to be built if $u < x$ or if $u = x$ and $v < y$.

One thing that is important for the king is that the K roads that will be repaired should be connecting all the cities in the kingdom, this is, there should be a path between any pair of cities using only the K roads, if he can accomplish this, it will demonstrate his willingness to repair the roads to all cities, even if not all roads in the kingdom are being repaired this time. He also wants the set of roads to be repaired to be the one with most need, given two sets of roads S and S' each with their roads sorted from highest to lowest needness the set S has more need to be repaired if S_i needness is greater than S'_i .

Can you help the king find the set of roads to repair?

Input

The first line of input contains three integers separated by space N $(1 \leq N \leq 100)$, M $(1 \leq M \leq \frac{N(N-1)}{2})$, K $(1 \leq K \leq 1000)$ representing the number of cities in the kingdom, the number of roads in the kingdom, and the number of roads that should be repaired, respectively. The following M lines contains two integer numbers separated by a space u_i and v_i, representing that the $i-th$ road connects cities with needness indexes u_i and v_i $(u_i < v_i)$.

Output

If a set of roads that satisfies the kings needs can be repaired print a line with exactly N integer numbers separated by a space where the $i - th$ number represents the number of roads to be repaired that connect city with needness i. If no such set exists print "Impossible".

Input example 1	**Output example 1**
5 10 10 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	4 4 4 4 4

Input example 2	**Output example 2**

`3 2 3`
`1 2`
`1 3`
	`Impossible`

Input example 3	**Output example 3**

`4 2 2`
`1 2`
`3 4`
	`Impossible`

Problem L

Let's run!

Two runners have been training for an important race in a 188888881-meter track, lets call them runner 1 and runner 2. They both train a lot; however, their performance is different. In the case of runner 1, the more training, the better the results, and their velocity can be expressed as $v_1 = D_t + E$. On the other hand, the velocity of runner 2 satisfies the equation $v_2 = At^2 + Bt + C$, which means in their training will reach their maximum capacity at some point, after it will go down (consider t the time they have spent training).

Knowing this, runner 2 decided to challenge runner 1 when the difference between their performance is maximum and, of course, when he is better than runner 1; in this way, he is sure he will win. Nevertheless, in the last minute, runner 1 discovered the crafty intentions of his rival, and although he will not be able to change the end of the race, wants to have a misleading photography where he is ahead of runner 2 for exactly 10 meters.

Then, runner 2 asked their best friend to take the photography at the right moment. Considering the aforementioned and the fact that our runners will take a rest before the race and this will cause their velocity takes the value of the biggest integer less than or equal to v_1 and v_2, respectively, how many minutes will have to lapse before the photography can be taken for the first time if we want to be sure both runners have run complete minutes, in case it is possible? Note: The race is over when runner 1 says it is over.

Input

The first line will have a single integer T, $1 \leq T \leq 10^4$, the number of test cases; then you will receive T lines, each of them with five real numbers A,B,C,D,E such that $1 \leq v_1, v_2 \leq 10^6$. Also, $A < 0, B, C, E \geq 0$ and $D > 0$.

Output

For each test case, print the minutes before runner 1 can make their photography. If it is not possible to get the photography, print -1.

Input example 1	Output example 1
2 -0.0000101 1.010101 0 0.33333 5000 -1 10 15 1 3	128350248 59027775

Problem M
MEOW

Some researchers have found cats have a language, this language is very well known by most cat lovers, all cats do this sound that is heard as a 'meow', this was some guess by cat owners, until today.

Researchers found that cats communicate using sounds similar to these four letters 'm', 'e', 'o', and 'w'. And have found that in cat language these sounds appear always in the same order, this is, if the cat will do an 'm' sound it will always appear before, 'e', 'o', 'w'. Some valid words in the cats language are 'mmmmeow', 'eow', 'e', 'w', " (silence), while 'wo', 'em', 'meeeewwoo' are invalid.

The research has also discovered that any string in the cats language always have between l_m and u_m sounds 'm', l_e and u_e sounds 'e', l_o and u_o sounds 'o', and l_w and u_w sounds 'w', it seems then that cats have much more words that we tought!.

We always thought Baker was a very talented cat, so we brought him to some examinations regarding the cats languages, researchers (and us) were shocked when we heard Baker did a sound 'owwwwmeeeeow', further investigation on baker showed that he is able to make much more words as other cats, it was found that if we take two words from the traditional cats language, Baker can make it sound as a single word.

Researchers are very busy studying the traditional cats language, but we want to study Baker's language and know how many different words Baker can make. Can you write a program that given the restrictions in the traditional cats language calculates the number of different words Baker can make?

Input

The first line of input contains a single integer number T ($1 \leq T \leq 100$) representing the number of test cases. T test cases follow, each test case consists of a single line with 8 integer numbers separated by a space, representing the values for l_m, u_m, l_e, u_e, l_o, u_o, l_w and u_w respectively. ($0 \leq l_m \leq u_m \leq 100$, $0 \leq l_e \leq u_e \leq 100$, $0 \leq l_o \leq u_o \leq 100$, and $0 \leq l_w \leq u_w \leq 100$)

Output

For each test case in the input print a line with a single integer, representing the number of different words Baker can make based on the traditional cats language restrictions.

Input example 1	Output example 1
3 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0	1 3 12

Problem N
New Pump System

The mexican water company uses a complex water pump system to clean water for the city. The company needs to replace a lot of old water pumps that have failures, they have removed the failing pumps from the map and left only the pumps that did not require a replacement.

Once the replacements were ready they found they did not know the place for each of the replacements as they missed to save the original map, that's why they need your help to find the way the pumps should be placed in the map.

The map is a square of size N, each cell can have maximum 1 pump, the pump system should follow some rules that should be followed carefully in order to avoid having problems with the resultant pump system:

There are 4 types of pumps (numbered from 1 to 4). A pump with number 1 needs to be connected with exactly 1 pump. A pump with number 3, needs to be connected with exactly 3 pumps, in general a pump with number n needs to be connected with exactly n pumps. Two pumps are connected if they are adjacent (share one side) in the map. If a pump does not meet this condition then the water will leak and won't be properly cleaned. For example, the following are valid configurations.

	1	
	3	1
	1	

	1	
1	3	1

	1	
1	3	
	1	

1	3	1
	1	

The pumps connections should be done carefully, if two connected pumps have the same number, they will explode. The following is an example of invalid connected pumps:

1	2	2	1

Your task is to complete the pump system, using the minimum water pumps and connecting every pump as described above. You should not move any of the working pumps (i.e the pumps given in the map).

Input

The input file contains several test cases, the first line contains an integer T $(1 \leq T \leq 15)$ the number of test casses. Following T test cases for each test case the first line contains an integer M $(1 \leq M \leq 8)$ indicating the size of the room system. The room is always a square. There are M lines following. Each line contains exactly M characters, a dot "." represents an empty space and a number (from 1 to 4) represents an already set pump. The input finishes on the end of file.

Output

For every test case you should output the full room, with every pump connected according to the rules. It is guaranteed that the solution exists and is unique.

Input example 1	Output example 1
2 5 .1... ....3 .4.4. 2.42. ..1.. 6 1...2. ...4.1 3.4... ...2.1 1.2... .1...2	.1..1 23.23 34342 2342. ..1.. 1.232. 2.3431 3243.. 2.32.1 1.2.23 .13232