Gran Premio de México 2019

Repechaje

Oct 26th, 2019

Contest Session

This problem set contains 12 problems; pages are numbered from 1 to 15. Without considering this nor the General information page.

This problem set is used in simultaneous contests:

Gran Premio de México 2019 – Repechaje
Gran Premio de Centro América 2019 – Repechaje

General information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp`, *codename*`.java`, *codename*`.py2` or *codename*`.py3`, where *codename* is the capital letter which identifies the problem.

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. There are no empty lines.
4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, has the usual end-of-line mark.

Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, must have the usual end-of-line mark.

Development team

The following persons helped to develop the problem set by creating and improving statements, solutions, test cases and input and output checkers:

Juan Pablo Marín,
Lina Rosales
María Celeste Ramírez Trujillo
Moroni Silverio, FES Acatlán
Roberto Solís, Universidad Autonóma de Zacatecas
Ariel Ortiz Ramírez, ITESM Campus Estado de México
Filiberto Fuentes Hernández, ESCOM
David Morales, ITSUR

Problem A – Automated Vehicle Search.

Author: David Morales

ITSURIN has invented a rails system which resembles a scale train, this allows him to store without any problem his priceless collection of 1/64 scale vehicles.

Even when the vehicles are at scale he has a lot of them, so the rails system takes a lot of space of ITSURIN house, this is why he usually takes a lot of time to find a vehicle. As the good engineer he is, ITSURIN decided to implement a mechanism using arduino, sensors, servomotors, and an integrated interface where he can input the ID of the vehicle he is looking for and then the system will find and retrieve such vehicle if it exists.

ITSURIN always stores the vehicles in the system by increasing order of ID, so he decided to use a pretty clever search algorithm to find the vehicles, his algorithm is as follows:

1. Let $S = int(sqrt(N))$ where N is the number of vehicles in ITSURIN's collection, and $int(X)$ is a function that returns the integer part of the real number X.
2. Let K be the input ID the mechanism will search for.
3. Put the ID scanner in position 0 just before the first vehicle.
4. While the ID read by the scanner is less than K and the scanner position is less than N, move S steps forward.
5. If the ID read by the scanner equals K. Retrieve the vehicle and stop.
6. Move S steps backwards.
7. While the ID read by the scanner is less than K, move 1 step forward.
8. If the ID read by the scanner equals K. Retrieve the vehicle and stop.

   ITSURIN has been running some experiments with his algorithm, and he believes it takes a lot of steps to find a given ID. He wants your help to calculate how many times the scanner will move before the mechanism stops.

Input

The first line of input contains two integer numbers separated by a space N ($1 \leq N \leq 10^6$) and Q ($1 \leq Q \leq 10^5$), representing the number of cars in ITSURIN's collection and the number of IDs ITSURIN will query. Each of the next N lines contains an integer number a_i ($1 \leq a_i \leq 10^9$ and $a_i < a_{i+1}$ for all i) representing the ID of the i-th positioned vehicle in ITSURIN's collection. Each of the following Q lines contains an integer number K_i ($1 \leq K_i \leq 10^9$), representing the ID of ITSURIN's i-th query.

Output

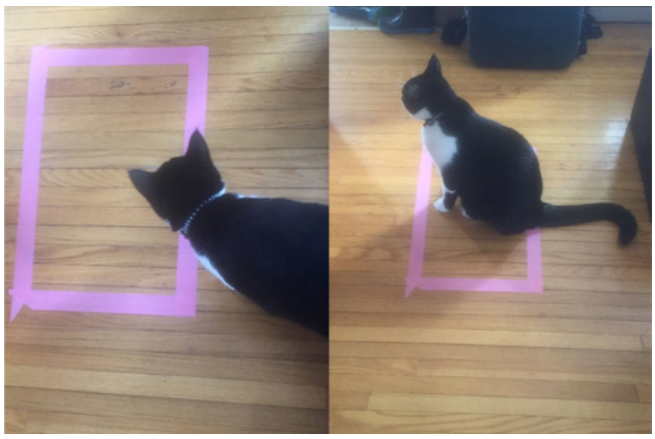
For each query in the input print a line with an integer number representing the number of times the scanner will move before the mechanism stops, if the mechanism will not stop with ITSURIN's algorithm print -1 for that query.

Sample input 1	**Sample output 1**
6 6 100 105 109 205 300 700 300 105 100 700 500 205	5 1 3 3 -1 2

Problem B – Baker Sleeping Again.

Author: Lina Rosales

Most cats love to sleep in well-defined spaces and Baker is no exception. As you know, he loves to sleep in rectangular-shaped places. His owner decided to create for him a rectangular shape in her room. To do this, She knows the coordinates x,y of the objects in her room, she will tie a piece of rope to different objects she has in the room to create rectangles with the rope.



Help Baker's owner to find the area of the biggest rectangle she can make using the objects in the room. Baker does not care if there are other things inside the rectangle, Baker can always squeeze in, as long as it is the biggest rectangle.

Input

The first line of input contains an integer T, the number of test cases. Each of the T test cases start with a line with an integer N ($4 \leq N \leq 50$), representing the number of objects in Baker's owner room, followed by N lines, each of these lines contains two integer numbers separated by a space x_i, and y_i ($0 \leq x_i, y_i \leq 1000$). representing the i-th object is in the coordinate (x_i, y_i) of the room. Every point (x_i, y_i) is unique.

Output

For each test case print "d" (without the quotes) where d is the maximum area with 9 decimal places. If no rectangles can be formed, output -1.000000000. Your answer value must be within 10^{-9} absolute error to be considered correct.

Sample input 1	**Sample output 1**
3 4 0 0 1 2 0 2 1 0 4 0 0 1 2 0 2 1 3 4 0 5 5 2 6 15 11 12	2.000000000 -1.000000000 68.000000000

Problem C – Calculate Taxes for Planet E-13.

Author: Roberto Solís

Planet E-13 orbits around a star in a far away galaxy named UAZ. In the calendar used in that planet, a year has 1000 days. Every year they have to pay taxes on their income. In order to avoid problems with the IRS (Internal Revenue Service), Thanitos, who lives on Planet E-13, keeps a ledger where he writes how much money he earned in the day for every day he was able to earn money. Your work is to help Thanitos calculate the total money he made in the year, as well as the tax rate and the tax he has to pay to the IRS, knowing that the currency on Planet E-13 has 5 decimal digits and that the maximum money he earns on a given day is $1000000000000000.00000

The tax to pay is calculated according to the following table:

Yearly Income Range	**Tax Rate**
$0.00000 - $10000000.00000	0%
$10000000.00001 - $100000000000.00000	10%
$100000000000.00001 - $1000000000000000.00000	20%
$1000000000000000.00001 - $10000000000000000.00000	30%
$10000000000000000.00001 - $100000000000000000.00000	40%

That is, if for instance, Thanitos earns $4006006000043.50408 in a year, he has to pay 20% of its earnings as tax, which amounts to $80120120008.70082

Note that the IRS is very touchy, and even for an error of 0.00001 in the tax calculation, Thanitos could get a large fine, therefore the sum and the tax calculation have to be accurate. Also, the rounding method to be used is round to even. In this method, the last digit after calculating the quantity he has to pay as tax, tells you to round up (if it is 6, 7, 8, or 9) or to round down (if it is a 1, 2, 3, or 4). When the value you intend to round off is a five, you MUST look at the next to last digit. If it is even, you round down. If it is odd, you round up. So, for instance, if Thanitos earned $20000041.00025 in the year, he would have to pay a 10% tax rate according to the table above. This means that we have to multiply $20000041.00025 by 0.1 to get the amount to pay, which is $2000004.100025. However, the last digit will not be stored since planet E-13 only uses 5 digits, and since the last digit is 5 and the next to last digit is 2, which is even, we would round down and the tax to pay would be $2000004.10002. If on the other hand, Thanitos had earned $20000041.00035 in the year, he would have to pay $2000004.10004 (same 10% tax rate, but since the last digit is 5 and the next to last digit is 3 when we calculate the 10%, we round up).

Input

The first line of input contains an integer N, representing the number of days Thanitos earned money during the year. Each of the following N lines contains an integer number representing the money Thanitos earned on a day of the year. Note that Thanitos does not have to earn money every day of the year, but assume that he earns money at least a day in the year. Each line has a number varying from 0.00001 to 100000000000000000.00000 and each line has always an integer part and a decimal part, with the decimal part always given with 5 digits.

Output

A single line containing values separated by a space: total, rate and tax, where total represents the total money Thanitos earned in the year (accurate to five decimal places), rate is the tax rate he has to pay (followed by a %) and tax is the amount he has to pay as tax to the IRS (also accurate to five decimal places).

Sample input 1

```
3
0.00030
10000000000000.16000
88575757.00265
```

Sample output 1

```
10000088575757.16295 30% 3000026572727.14888
```

Sample input 2

```
3
891412.18091
5000000000000.09872
4999999642341.04062
```

Sample output 2

```
10000000533753.32025 30% 3000000160125.99608
```

Problem D – Distributing Pizza.

Author: Juan Pablo Marín

Site manager of El Gran Premio bought a pizza to be distributed betweeen the N contestants of the contest site, the pizza is a classical circular shaped pizza with chesse and pepperoni, it has K slices, and as you may guess the pepperoni is not distributed evenly into the pizza slices.

To avoid this being a mess, the Site manager decided to make exactly N cuts in the pizza to divide it in N parts, each of the contestants will take one of the parts and therefore they will only need to take pizza one time, minimizing the odds for someone to throw the pizza to the floor. Contestants in the site love pizza, but, they care more for the manager to be fair. The contestants defined the "unfairness" of the pizza distribution between contestants i and j as the difference in the amount of pepperoni that exists between the pizza parts that contestants i and j took.

The site manager wants to avoid "unfairness" as much as contestants, that is why he has come to you as the most experienced programmer and asked to create a program that given the amount of pepperoni each slice of the pizza has, determines, what is the way to cut the pizza in order that the maximum "unfairness" between any pair of contestants is as small as possible.

Input

The first line of input contains two integer numbers separated by a space N and K ($1 \leq N \leq K \leq 100$). Each of the next K lines contains an integer number the i-th line represents the amount of pepperoni the i-th slice in the pizza has, no slice will have more than 10^5 pepperonis.

Output

Output a single line with an integer representing the maximum "unfairness" between any pair of contestants in the pizza distribution where this value is the smallest.

Sample input 1	**Sample output 1**
1 4 5 10 15 20	0

Sample input 2	**Sample output 2**
2 4 5 10 15 21	1

Problem E – Entertainment With Prefixes.

Author: María Celeste Ramírez.

Santiago is a curious boy, he has been studying languages and properties of the words, he took the dictionary the other day and very excited started to explain to you what a a suffix is, what a prefix is, and why a lot of consecutive words in the dictionary seem to have a lot in common when written but have completely different meanings.

Being aware of what a prefix is, you decided to put an easy task for Santiago, you write a list of words in a page of his notebook and ask him to find the number of words that are prefixes of another word in the page, you are surprised to see how fast he finds all such words. As he really likes challenges he then challenged you with these words: "It was easy, since you are older you need a bigger challenge, can you find a subset of words from the list such that no word in that set is a prefix of another word in the same set?". You decided to take the challenge, since it seemed so easy, given that you can form a subset with only one word and that way, it is not a prefix of any another word in the set, but, you want to impress Santiago, and to impress him you decided to write a program to find the largest possible subset.

Write a program that given the list of words, it finds the largest possible subset that holds Santiago's constraints.

Input

The first line of input contains an integer N ($1 \leq N \leq 10^4$), the number of words in the list. Each of the next N lines contains a string consisting of up to 100 lowercase english alphabet characters.

Output

Output a single line with an integer indicating the length of the largest possible subset that holds Santiago's constraints.

Sample input 1	**Sample output 1**
3 ab ac a	2

Sample input 2	**Sample output 2**
5 aa ab ac ad ae	5

Problem F – Filiberto's Party.

Author: Filiberto Fuentes

This year Filiberto offered a party due to his birthday in June. This party was intended to have a different format from the ones he has organized before. In this event there were a lot of people wishing to play some videogames that works only with a WIFI network.

The router of the WIFI network have a distance limitation, which means that the internet will be available only to at most a distance d from the router (to any direction). There were some problems with the WIFI connection during the party and that's why Filiberto wants you to help him with this problem.

He will organize an event next year, but he wants to avoid this kind of problems. The party will be held in an apartment of size nxm (the apartment is represented as a matrix of this size). In order to play the videogames, every person will be seated in a point p in the apartment **(No two persons will be seated in the same point** p**)**.

Filiberto does not know which people will attend the party, so he is going to ask you several queries of the following form: $k\ a_1\ a_2\ a_3\ ...\ a_k$ - which represents a query with a number k, meaning that he is going to ask you for k people and then the identifiers of each person (people's identifiers goes from 1 to l).

What you need to print after each query, is the minimum distance d that the router should support in order to let all people in the current query play without any issues.

You can put the router in any position (can be real values coordinates) inside the apartment. You can also put it in the same position where a person will be located.

Input

The first line of input contains three integers n, m and l ($1 \leq n, m \leq 10^9, 1 \leq l \leq 18$) - representing the size of the apartment and the maximum number of people that could attend the party.

Next l lines contain two integers x_i and y_i ($1 \leq x_i \leq n, 1 \leq y_i \leq m$) - representing the point p where the i-th person will be located.

Next line contains an integer q ($1 \leq q \leq 3 * 10^5$) - the number of queries that Filiberto is going to ask you. Next q lines contain each query with the following format: $k\ a_1\ a_2\ a_3\ ...\ a_k$ ($1 \leq k \leq l, 1 \leq ai \leq l \forall i | 1 \leq i \leq k$)- which represents a query with a number k,meaning that he is going to ask you for k people and then the identifiers of each person. In each query each person will appear at most once.

Output

For each query you need to print the minimum distance d that the router of the WIFI network should support considering that you can put the router in any position in the apartment.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-9}

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{max(1,|b|)} \leq 10^{-9}$.

Sample input 1	**Sample output 1**
10 10 4 10 5 3 6 9 7 4 4 5 1 4 2 1 4 3 1 2 3 2 2 4 3 1 2 4	0.000000000 3.041381265 3.535533906 1.118033989 3.535533906

Problem G – Guarding The Museum.

Author: Moroni Silverio

Jaime has opened a new museum with lots of exhibitions in it, some of them very expensive and beautiful to be honest. Jaime loves order not chaos, so he was involved in the creation of the museum blueprints. He did that because he wanted to ensure visitors don't get lost while visiting the museum and for that reason, there is only one way to get from one exhibit to another going only once trough some intermediary exhibits.

Jaime's museum admission is free for everyone, however, someone has to pay for the security, these fancy painting won't take care of themselves. In order to avoid having the paintings stolen a security company, called ICPC (International Corporation that Protects Cows) was hired which charges per the number of security guards assigned to the museum.

Museum's budget is based upon charity, and it is limited so Jaime wants to employ only necessary personnel to guard the museum taking into account every exhibit in the museum must be protected by some guard.

According to ICPC rules, guards can only be placed in exhibit rooms, and guards can protect exhibit rooms connected trough a hallway with the exhibit room they are placed in, including it.

Help Jaime save some money by telling him the minimum amount of guards required to protect each exhibit. Don't worry, the blueprints of the museum will be given.

Input

The first line of the input contains two numbers N ($1 \leq N \leq 10^5$) y C ($1 \leq C \leq 10^5$) the number of exhibit rooms in the museum and the cost per guard. After this $N-1$ lines will follow, each with a pair of numbers a, b ($1 \leq a, b \leq n$) representing that exhibit room a and exhibit room b are connected by a hallway.

Output

Output the minimum cost required to protect the whole museum

Sample input 1	**Sample output 1**
4 1 3 2 1 3 4 3	1

Problem H – Hidden Lists.

Author: Juan Pablo Marín

It is relatively easy to create big numbers starting with a list of integer values, suppose you have the list $[1, 2, 3, 5, 123]$ if you concatenate all the numbers in the list you will get the number 1235123. This is an easy method, but, remember that, if a number has an arbitrary amount of leading zeroes, the number has the same value and therefore we could say having the list $[1, 02, 3, 5, 123]$ is considered the same list as before (the values are the same and in the same order, even when the second element is 02). As you can see then, we can create a lot of numbers using the list of values $[1, 2, 3, 5, 123]$ as starting list, for example 01235123, 01000200003050123, etc...

Can you make a program that finds how many distinct lists of increasing values with values between 1 and 10^6 exists that can generate a given number N? A list of increasing values is a list such that no value is less than the previous values in the list. Two lists are considered to be different if they are of different length or if their values differ at some position.

Input

The input consists of a single line that contains the number N ($1 \leq N \leq 10^{60}$), representing the number to be generated.

Output

Output a single line with an integer indicating the number of distinct lists of increasing values that can generate the number N.

Sample input 1	**Sample output 1**
`00001`	`1`

Sample input 2	**Sample output 2**
`123`	`3`

Sample input 3	**Sample output 3**
`10000000`	`0`

Problem I – Invoice Cost Reduction.

Author: Moroni Silverio

Jaime has been learning web development for a while, sometimes he loves it, sometimes he hates it. Recently he learned about a technology called serverless computing in which he doesn't have to pay for a server running 24/7 which is not being used every hour of the day.

This technology has the concept of "pay as you go", which means he only pays every time a section of code is executed, if nothing is executed he doesn't have to pay anything.

According to the invoices he receives from the serverless company which hosts his code, he is being billed for the execution time of his code.

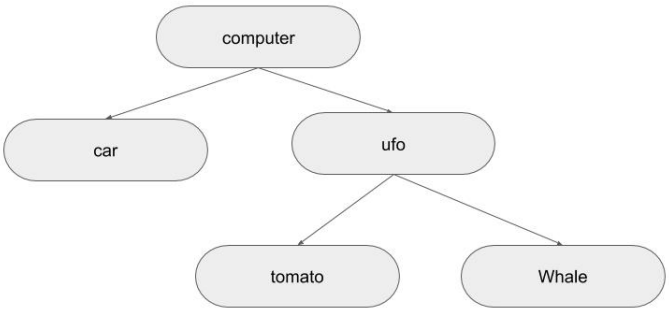
Jaime has modernized his little store and now he is selling via the internet and has implemented a function that retrieves the prices of the products he is selling.

Originally he had the product's prices in a list, but finding their prices was time consuming. So he decided to store the prices in a binary search tree.

A binary search tree goes as follows: Give a node in the tree (root for example), the value stored in it [the name of a product] is lexicographically greater than any other value of any other node of the left sub tree and that same value is lexicographically smaller than any other value of any other node of the right sub tree.

The execution time of the function that retrieves the price of a product depends on the number of comparisons made in order to find the product in the tree.

For example, if the products for sale are *computer*, *car* ufo, *tomato* and *whale* and the following tree is made:



Two comparisons would be made to find the ufo's price, three comparisons to find tomato's price, and only one comparison to find the computer's price.

Jaime also loves data mining and has stored the frequencies in which the price of each product is asked. Jaime has created a function to calculate the expected invoice to be paid and is defined as follows:

$$\sum_{product \in Products} Frequecy(product) * Comparisons\_to\_find(product)$$

Jaime has discovered that different binary search trees may lead to different expected invoices. Can you help him find the best tree that minimizes the expected invoice? Of course you can.

Input

Input begins with a number N ($1 \leq N \leq 100$), the number of products Jaime sells over the Internet, N lines will follow, each describing a product for sale in the following manner: a nonempty string s representing the name of the product (the name is composed of at most 100 characters in lower case $[a - z]$ and no two product have the same name), then, in the same line two integer numbers p and f, ($1 \leq p \leq 10^5$) and ($1 \leq f \leq 10^7$), the price and the frequency of the mentioned product.

Output

Output the minimum expected invoice achievable.

Sample input 1	**Sample output 1**
5 computer 50 50 car 800 40 ufo 90 40 tomato 1 30 whale 100 30	390

Problem J – Jaime's Internet Sale.

Author: Moroni Silverio

Jaime, the lovely guinea pig who loves programming, is about to move to another city and he wants to sell some old items (all functioning by the way) because the moving truck may be expensive and out of his budget. Jaime has N items for sale.

To get rid of his stuff as soon as possible he posted on the Internet everything he is selling along with the money he is expecting to receive for each item.

Three of his closest friends(Jiren, Ricardo and Sarai) saw the ads and offered money for each article Jaime is selling. Jaime will decide which friend he will sell each item. It is not mandatory Jaime sells all his items to the same person.

However, Jaime's best friend is Jiren and he has asked Jaime a special favor: that Jaime sells him exactly J items, why? We don't know. As Jaime is a good friend, he has agreed with Jiren's petition.

Nevertheless, Jaime wants to maximize his profits, if we can call them that, without breaking his word of selling Jiren exactly J items.

Can you help Jaime to tell in which way he can sell his items so that his profits are as much as possible?

Input

The first line of the input contains two integers N ($1 \leq N \leq 2 \cdot 10^5$) and J ($0 \leq J \leq N$) representing the number of items Jaime put for sale (maybe he shouldn't have accumulated so much stuff over the years) and the number of items he must sell to Jiren. After that, four lines will follow each with N integer numbers $0 < a_{ij} < 10^5$, with $i \in 1, 2, 3, 4$ and $j \in 1, 2, \ldots, N$.

The first of these four lines describe what Jaime is expecting to receive for each item he is selling (a_{1j} represents what Jaime wants for the $j - th$ item)

The next three lines represent what each of his friends (Jiren, Ricardo and Sarai respectively) are offering for each item in the same format as the first of the four lines.

Output

Output the maximum amount of money Jaime can get by selling all of his items to his friends, if the amount is less than the total amount of money he expects to receive output "Don't do it, buddy".

Sample input 1	**Sample output 1**
4 1 1 2 3 4 5 1 1 1 1 1 1 1 1 2 2 2	11

Sample input 2	**Sample output 2**
1 1 10 9 8 7	Don't do it, buddy

Problem K – Keep Branches Ordered.

Author: Juan Pablo Marín

Nlogonia postal offices have N branches across the cities labeled with numbers from 1 to N, to deliver the mail each morning buses go from one branch to another based on a map of branches that has been created. This map of branches specify to what branches a bus that depart from branch i can go next in his traveling. Since the buses leave early morning, it is very common that some bus drivers do not pay attention to the branches map and go to branches not intended for the branch they depart from, this is a problem for the postal offices deliveries as people that expect their packages to be delivered at a branch may not get them in time.

The postal offices manager has determined that a relabeling of the branches should be made so that if a bus departs from a branch with the label i, any of its destinations branches j should have a label greater than i, the manager believes this will help stop bus drivers going to the wrong branches because now they can at least check if they are driving to a branch with a higher label than the one they depart from.

Given the branches map, help the postal offices manager determine what the new label for each branch should be.

Input

The first line of the input contains an integer N ($1 \leq N \leq 100$), representing the number of branches. Each of the next N lines contains N integer numbers separated by a space, the j-th integer in the i-th line represents whether or not bus drivers should go from branch i to branch j based on the branches map.

Output

Output a single line with N integers separated by a space where the i-th integer represents the new label for the i-th branch, or -1 if no possible relabeling exists. If more than one labeling exists, print the lexicographically smallest. For two labelings p and q, we say that p is lexicographically smaller than q if and only if there exists an index $1 \leq l \leq N$ such that: For any $1 \leq i < l$, $p_i = q_i$ and $p_l < q_l$.

Sample input 1	**Sample output 1**
4 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0	1 2 3 4

Sample input 2	**Sample output 2**
3 0 1 0 0 0 1 1 0 0	-1

Problem L – Lisp to Infix Translator.

Author: Ariel Ortiz Ramírez

Your good friend Progra María de la Codificación is taking an advanced undergraduate class calleda *Intriguing Computer Programming Curiosities* (ICPC) where she is learning Lisp (acronym for *List Processor*), a fairly ancient but highly respectable and intriguing programming language.

Lisp programs are built using a tree-structured nested list notation called *s-expressions* (symbolic expressions). For our purpose, an *s-expression* can be recursively defined as:

1. An *atom*, which is a single capital English letter from 'A' to 'Z'.
2. A *list* of the form '(op s1 s2 ...)' where 'op' is either the addition '+' or multiplication '*' operator, and 's1 s2 ...' are two or more *s-expressions* that serve as operands. A single space character is used to delimit the elements contained within the list (for example, to separate 'op' from 's1'). There are no spaces after an opening parenthesis '(' nor before a closing parenthesis ')'.

The following are examples of valid *s-expressions*:

- A
- (+ A B C)
- (* D D D D)
- (+ A B (* C (+ D (+ E F) G H) (* I J)) K L)

Given that '+' and '*' represent prefix operators, *s-expressions* can be converted into the ordinary infix notation that we typically use in other more conventional programming languages like C or Java. Thus, the previous examples are equivalent to:

- A
- A+B+C
- D*D*D*D
- A+B+C*(D+E+F+G+H)*I*J+K+L

As can be observed from the last example, we avoid using more parenthesis than those that are strictly required to preserve the semantics of the original *s-expression*. Remember that multiplication has a higher precedence than addition when using the ordinary infix notation.

Progra María is having trouble understanding programs written in Lisp. Because you want to help your friend, you have decided to write a program that automatically translates Lisp *s-expressions* into ordinary infix notation.

Input

The first line of the input contains an integer N that specifies the number of test cases that follow ($1 \leq N \leq 100$). Each of the following N lines contains a valid *s-expression* as previously described. The size of each individual line is no greater than 15000 characters. Also, lists can be deeply nested, but no more than 350 levels.

Output

For each test case, output in a line the translated expression in ordinary infix notation. Please note that the results should not contain spaces anywhere.

Sample input 1

```
4
A
(+ A B C)
(* D D D D)
(+ A B (* C (+ D (+ E F) G H) (* I J)) K L)
```

Sample output 1

```
A
A+B+C
D*D*D*D
A+B+C*(D+E+F+G+H)*I*J+K+L
```

Sample input 2

```
3
(* (+ N O P) (+ (+ (* (+ Q R) (+ S T)) U) (* (+ V W X) (+ Y Z))))
(+ T (* T (+ T (* T (+ T (* T (+ T T)))))))
(+ (* (+ (* (+ (* (+ T T) T) T) T) T) T) T)
```

Sample output 2

```
(N+O+P)*((Q+R)*(S+T)+U+(V+W+X)*(Y+Z))
T+T*(T+T*(T+T*(T+T)))
(((T+T)*T+T)*T+T)*T+T
```