Primera Fecha Gran Premio México 2018

Apr 28th 2018

Problems book

General information

This book contains a total of 10 problems; Pages are numbered from 1 to 12 not including this page. Verify your book is complete.

Problem A. Permuting and adding up

Source file name:	`A.c, A.cpp, A.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Óscar Dávalos`

Given a positive integer number N, your task is to find how many different numbers exist that have the same digits as N and the sum of all these numbers.

As an example, if $N = 120$ then there are 6 numbers : 012, 021, 102, 120, 201 and 210. The sum of these numbers is $12 + 21 + 102 + 120 + 201 + 210 = 666$.

Input

The first line of input contains a single number T the number of test cases. The next T lines contain a single positive integer number N.

- $1 \leq T \leq 10^4$
- $0 \leq N \leq 10^{11}$

Output

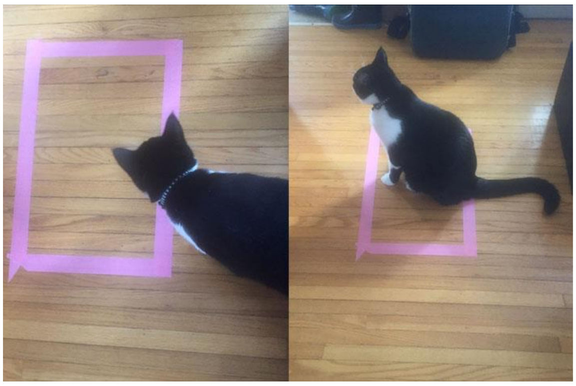
For each test case your program should print a single line with two integers separated by space. The first number represents the number of different numbers that exist with the same digits as N, the second number contains the sum of all these numbers.

Example

Input	Output
2 120 112	6 666 3 444

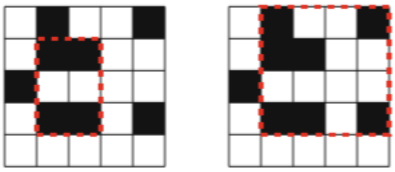
Problem B. Sleeping Baker

Source file name:	`B.c, B.cpp, B.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Felipe Baquero, Lina Rosales`



Most cats love to sleep in well defined spots and Baker is not the exception, he loves to sleep in rectangular shape places. His owner decided to create for him a rectangular shape in her room. To do this, she will tie a piece of rope to different objects she has in the room to create rectangles with the rope.

In the following figure, the black sections are places in the room where the owner has an object where she can tie the piece of rope to create the rectangles. The red lines show possible rectangles she can create.



Help Baker's owner to discover how many different rectangles she can make for her beloved cat.

Input

There are multiple test cases. Each test case starts with a line with two integers separated by a space n and m describing the size of the room. The next n lines contain m characters each with the value 1 or 0, where 1 means that there is an object in that position were the rope can be tied of. The input ends at the end of file.

- $1 \leq n, m \leq 2500$

Output

For each test case print "case y: d" (without the quotes) where y is the number of test case starting with 1 and d is the number of different rectangles Baker's owner can make for him to sleep.

Example

Input	Output
5 5 01001 01100 10000 01101 00000 2 2 11 11 5 5 11111 11111 11111 11111 11111	case 1: 2 case 2: 1 case 3: 100

Problem C. Counting weak RNA sequences

Source file name:	`C.c, C.cpp, C.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Araceli Velázquez`

An interesting problem in the production of proteins using bioengineering is to be able to determine when a protein can be efficiently created by a micro organism and when it is not. Some hypothesis lead that a protein will have low production levels when the transfer RNA sequence section has several bases of adenine (A) and thymine (T) chained together. This due to the fact that adenine is chained to thymine by 2 hydrogen links, while cytosine (C) is chained to guanine (G) by 3 links.

A section of the RNA sequence is any substring from the RNA sequence, this is, is formed of consecutive RNA bases from the RNA sequence. For example in the RNA sequence 'ATCTC', 'ATC' is a section of the RNA sequence while 'ACC' is not as these chars are not consecutive in the RNA sequence.

A complementary section of a RNA sequence exists only on those sequences that have only two types of bases and is defined as the sequence that contains the contrary base in the same position, examples of complementary RNA sequences are: 'TAAAA' and 'ATTTT', 'TATATA' and 'ATATAT'.

Normal ending transcription sections usually contain only bases of adenine and thymine, however it is not that usual that the RNA sequence contain also the complementary sequence of an ending transcription section, this is why we consider a weak RNA sequence as a section of 10 elements from the RNA sequence that contains only adenine and thymine and that its complementary section exists in the given RNA sequence.

Given a RNA sequence your task is to find how many weak RNA sequences exists in it.

Input

The first line of input contains a single number T the number of test cases to follow. The next T lines contain each a RNA sequence, the RNA sequence contains only the characters 'A' for adenine, 'C' for cytosine, 'T' for thymine, and 'G' for guanine and it's length is no longer than 10^6.

- $1 \leq T \leq 10$

Output

For each test case print in one line the number of weak RNA sequences that exist in the given RNA sequence.

Example

Input	Output

`2`
`ATATATATATA`
`GGCAAAGATATCGATCG`
	`2`
	`0`

Explanation

In the first test case there are two weak sequences: 'atatatatat' and 'tatatatata'. For the second test case there are no weak sequences.

Problem D. Dividing Hexadecimal Numbers

Source file name:	`D.c, D.cpp, D.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas Hernández`

Mr. Homft promised not to let homework to the kids that could accomplish a task in class. Today's lecture was about hex numbers. Hex numbers were invented by the lazy computer scientists from the last century who didn't want to write a lot of zeros and ones, so they synthesized binary into hex numbers. For a binary number it's possible to form groups of 4 bits and replace them by the hex digit. An easy way to convert from hex to binary! A hex digit is represented as a number from 0 to 9 or an uppercase letter from A to F which represents numbers from 10 to 15.

Mr. Homft's class is integrated only by smart kids, so don't get scared by their abilities. The last month they were studying Newton's laws, number theory and some other kind of sorcery tricks. Well, now that everything has been explained let's go to Mr. Homft's problem. He wrote a hex number N , actually a really huge one, and then he asked if the number was a multiple of 17. Passed no more than 5 minutes, all the kids had answered to the question correctly. You may be wondering if you must have taken the sorcery course last summer to calculate the result but maybe what you need is a different course. Can you answer correctly as the kids? Given a set of hex numbers your task is to say whether they are or not multiples of 17.

Input

You'll have to read until end of file. Each line of input will have the hex number N .

- N will have at most 10^5 hex digits.

Output

For each line of input, write "yes" if the hex number is divisible by 17, write "no" otherwise

Example

Input	Output
9999 11 AA 0 1	yes yes yes yes no

Problem E. Exact sum of squares

Source file name:	E.c, E.cpp, E.java
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marín

Given the prime factors of a number N, can you find if there are two numbers a, b such that $a^2 + b^2 = N$?

Input

The first line of input contains a number K the number of prime numbers in the primer factorization of N. Each of the next K lines contain two numbers (P_i, B_i) separated by a space, the first number (P_i) is a prime number in the prime factorization of N the second number is the number of times that P_i appears in the prime factorization of N.

- $1 \leq K \leq 100$
- $1 \leq P_i \leq 10^6$
- $1 \leq B_i \leq 100$
- You can assume all values for P_i are prime numbers and none of the primer numbers repeat in the input.

Output

For each test case you must print a line with the string "Yes." if there are two numbers a, b such that $a^2 + b^2 = N$, print "No." otherwise.

Example

Input	Output
4 2 1 3 4 5 1 7 2	Yes.

Input	Output
2 3 3 11 4	No.

Explanation

In the first test case the given number $2^1 * 3^4 * 5^1 * 7^2 = 39690 = 189^2 + 63^2$. For the second test case $3^3 * 11^4 = 395307$ can not be represented as the sum of two squares.

Problem F. Finding the train

Source file name:	`F.c, F.cpp, F.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Araceli Velázquez`

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.

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, the number of trains to test. The following T lines contain a string S 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.

- $1 \leq T \leq 10$
- $1 \leq |S| \leq 10^6$

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.

Example

Input	Output
3 ++-++++-++- +-++-+ -++	10 6 2

Problem G. Gambusines

Source file name:	`G.c, G.cpp, G.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Sergio Ivvan Valdez`

A gambusine explores a zone of R x C acres to determine if they can be exploded or not. The explosions are made on deposits that are a set of acres that are exploitable and contiguous, each acre will generate a benefit of g units when exploded, and the explosion of a deposit will take a cost of Txg regardless of it's size. A deposit is then profitable if it contains at least T exploitable and contiguous acres as the total benefit of exploding the sub area will be $(n_y - T)g$ where n_y is the number of acres in the exploitable sub area.

Given the rectangular map of R x C acres of the zone the gambusine will explore. You need to find the number of deposits that are profitable and the total benefit of exploding such deposits. The exploration zone is defined by a matriz M, such that $0 \leq M_{i,j} \leq 100$. If $M_{i,j} = 1$ then the acre (i, j) is exploitable. A deposit is a set of acres $M_{i,j}$ such that $M_{i,j} = 1$ and at least one of the acres in it's neighborhood $V_{i,j} = M_{i-1,j}, M_{i,j-1}, M_{i+1,j}, M_{i-1,j}$ is also exploitable, for any (i, j) in the deposit.

Input

The input file contains in the first line two values T and g. The next line will contain R and C the number of rows and number of columns respectively for the zone that the gambusine will explore. The next R rows contains C numbers separated by space representing the values for the zone as explained above.

- $1 \leq R, C \leq 100$
- $2 \leq T \leq R * C$
- $1 \leq g \leq 100$
- $0 \leq M_{i,j} \leq 100$

Output

The output file contains a single line with two numbers separated by a space, the first one is the number of profitable deposits in the zone to be explored, the second one is the total benefit after exploding such deposits.

Example

Input	Output
4 2 10 7 82 83 84 85 86 87 88 83 84 85 1 1 88 1 1 85 86 87 88 1 1 85 1 87 88 1 90 91 86 1 1 89 1 91 92 87 1 89 1 1 92 93 1 1 90 91 92 93 94 89 90 1 92 1 94 95 90 91 92 93 94 95 96 91 1 93 1 95 1 1	2 4

Problem H. Harder Sokoban

Source file name:	`H.c, H.cpp, H.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Félix Arreola Rodríguez`

Toby is a terrible person, he likes to steal games from internet to publish them as own. So, for his next robbery Toby found a game called "Sokoban", a game where a player pushes boxes or crates around in a warehouse, trying to get them to storage locations. It doesn't matter which box ends in which storage location, the only requirement is that all boxes final positions must be a storage location. Also, the boxes can "pass" through storage locations freely as they transit to another storage location. Below is a example of the classical Sokoban game:

		#	#	#			
		#	G	#			
		#		#	#	#	#
#	#	#	B		B	G	#
#	G		B	T	#	#	#
#	#	#	#	B	#		
			#	G	#		
			#	#	#		

Where '#' represents a wall (non-walkable), 'G' represents a storage location, 'B' represents a box, and 'T' represents the initial position of the player. The game itself is pretty difficult, because the player cannot pass through boxes (or walls), and only can push a box from his position, because of this, sometimes, an "easy" looking puzzle can be very tricky, like this one:

#	#	#	#	#	
#		T		#	
#	G	G	G	#	
#	B	B	B	#	#
#					#
#					#
#	#	#	#	#	#

Yes, indeed, this puzzle could be solved in only 3 moves if the player starting position was "below" the boxes. And because of this, Toby wants to "clone" this game in an different version, changing one simple rule: "The player can transport himself to any empty location". In this way, Toby can publish his "Harder Sokoban" game version.

Your task is help Toby to write a program to calculate the minimum number of box movements that are needed to put every box from his starting positions to a storage location (can be any storage location). Remember, two boxes cannot be pushed at the same time, and two boxes can't occupy the same space, a box can not be pushed through or over a wall or over another box.

Input

The first line contains two integer N and M indicating the size of the warehouse. The following N lines contains exactly M chars, a '#' represents a wall, a '.' (dot) represents a walkable tile, 'G' represents a storage location, 'B' represents a box. The map is always delimited by walls.

- $3 \leq N, M \leq 10$ There will be no more than 3 boxes in the input.

Output

You should output a single line with a integer with the minimum number of box movements to complete the level.

Example

Input	Output
7 6 ###### #...## #GGG## #BBB## #....# #....# ######	3

Problem I. Is a triangle

Source file name:	`I.c, I.cpp, I.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marin - CUCEI Guadalajara`

JP has N coins in his pocket. While he waits for the contest to start he puts 1 coin in the table, then he puts 2 coins below the one he put before and makes a triangle. Then he does the following:

- Let K be the last amount of coins JP put and $K' = K + 1$. JP Adds K' coins in such way the coins are still arranged in a triangle.
- Repeat the previous step until JP can not add more coins

If at the end of the procedure JP has no coins then it is said that N is a triangular number as all the coins can be arranged in this way to create a triangle.

Your task is to find given the number N if it is a triangular number, in such case you need to find the last value of K' JP used to create the triangle.

Input

The first line of input contains a number T the number of test cases. Each of the next T lines contain a single number N.

- $1 \leq T \leq 100$
- $3 \leq N \leq 10^{18}$

Output

For each test case you must print the value of the last K' used by JP to create the triangle given that N is triangular, print -1 otherwise.

Example

Input	Output
3 55 90 10	10 -1 4

Problem J. Joining cells

Source file name:	`J.c, J.cpp, J.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín - CUCEI México`

In Quadradonia as you can imagine everything is a square, all their buildings, all the streets, all the tiles, even the people are squares. In the recent years people have started to arrange their square energy cells in a bigger square of size N in such way that you can maximize the amount of energy that you can take from all the energy cells in town.

An energy cell in Quadradonia allows to put some machine over it and then the machine gets the amount of energy that the energy cell has. The new arrangement Quadradonians are doing allows to put a machine that expands over more than one energy cell and then the machine will get the energy as the sum of all the cells it is overlapping, if the energy a machine gets is greater or equal to it's minimum energy requirement then the machine will power on.

To test the new arrangement a new machine with a minimum energy requirement of X will be built, this machine will have also a square shape, can you find all the different ways you could put a machine in the energy cells arrangements such that the machine will be able to power on?

Input

The first line of input contains two integers separated by a space N and X. The next N lines contain each N integers separated by space representing the amount of energy the cell in that position brings.

- $1 \leq N \leq 1000$
- $1 \leq X \leq 10^9$
- The value of each energy cell $C_{i,j}$ is in the range $1 \leq C_{i,j} \leq 100$

Output

Your program should print a line with an integer number. The number of ways you can put a machine with X minimum energy requirement to be able to power on into the energy cells arrangement.

Example

Input	Output
2 4 3 4 3 2	2

Explanation

In the example test case there you are asked to find the ways to put a machine that requires 4 of energy to power on. There are two ways to put such machine: One is to use all the 2x2 square, the other is to put the machine in the 1x1 square represented by the upper right cell which sums 4 and is the required to power on the machine.