RIT Coding Questions-

Question Number 1

You are given a binary matrix (0s and 1s). Your task is to find the size of the largest square sub-matrix containing only 1s.

Input:

The first line contains two integers N and M ($1 \le N, M \le 100$), separated by a space. N is the number of rows, and M is the number of columns in the matrix.

The next N lines each contain M integers, either 0 or 1, representing the elements of the matrix.

Output:

Output a single integer, the size of the largest square sub-matrix containing only 1s.

Example:

Input:

```
5 5
1 0 1 1 1
1 0 1 1 1
1 1 1 1 1
1 0 0 1 0
```

Output:

3

Explanation: The largest square sub-matrix containing only 1s has a size of 3x3, starting at the top-left corner.

Test Cases:

1) Input:

```
4 4
1 0 1 0
1 0 1 1
1 1 1 1
1 0 0 0
```

Expected Output:

2

2) Input:

```
3 3
1 1 1
0 1 1
1 1 0
```

Expected Output:

2

3) Input:

5 6
0 0 0 1 1 1
0 0 0 0 1 1
0 0 0 0 1 1
0 0 0 0 0 0

Expected Output:

2

4) Input:

2 2
0 0
0 0

Expected Output:

0

Question No-2

Given an array arr of distinct elements of size N, the task is to rearrange the elements of the array in a zig-zag fashion so that the converted array should be in the below form:

arr[0] < arr[1] > arr[2] < arr[3] > arr[4] < . . . . arr[n-2] < arr[n-1] > arr[n].

Example 1:

Input:
N = 7
Arr[] = {4, 3, 7, 8, 6, 2, 1}
Output: 3 7 4 8 2 6 1
Explanation: 3 < 7 > 4 < 8 > 2 < 6 > 1
Example 2:

Input:
N = 4
Arr[] = {1, 4, 3, 2}
Output: 1 4 2 3
Explanation: 1 < 4 > 2 < 3

Constraints:
1 <= N <= 106
0 <= Arri <= 109
You cannot use temporary extra arrays. You have to rearrange the elements of same array.

Question Number- 3

Problem Description

Question -: A positive integer d is said to be a factor of another positive integer N if when N is divided by d, the remainder obtained is zero. For example, for number 12, there are 6 factors 1, 2, 3, 4, 6, 12. Every positive integer k has at least two factors, 1 and the number k itself.Given two positive integers N and k, write a program to print the kth largest factor of N.

Input Format: The input is a comma-separated list of positive integer pairs (N, k).

Output Format: The kth highest factor of N. If N does not have k factors, the output should be 1.

Constraints:

1<N<10000000000
1<k<600.
You can assume that N will have no prime factors which are larger than 13.

Example 1

Input: 12,3
Output: 4
Explanation: N is 12, k is 3. The factors of 12 are (1,2,3,4,6,12). The highest factor is 12 and the third largest factor is 4. The output must be 4.

Example 2

Input: 30,9
Output: 1
Explanation: N is 30, k is 9. The factors of 30 are (1,2,3,5,6,10,15,30). There are only 8 factors. As k is more than the number of factors, the output is 1.

Question- 4

Problem Description

One person hands over the list of digits to Mr. String, But Mr. String understands only strings. Within strings also he understands only vowels. Mr. String needs your help to find the total number of pairs which add up to a certain digit D.The rules to calculate digit D are as follow :-

Take all digits and convert them into their textual representation.

Next, sum up the number of vowels i.e. {a, e, i, o, u} from all textual representation.

This sum is digit D

Now, once digit D is known find out all unordered pairs of numbers in input whose sum is equal to D. Refer example section for better understanding.

Constraints

1 <= N <= 100

1 <= value of each element in second line of input <= 100

Number 100, if and when it appears in input should be converted to textual representation as hundred and not as one hundred. Hence number of vowels in number 100 should be 2 and not 4

Input

First line contains an integer N which represents number of elements to be processed as input

Second line contains N numbers separated by space

Output

Lower case representation of textual representation of number of pairs in input that sum up to digit D

Note: – (If the count exceeds 100 print “greater 100”)

Examples

Input : 5

1 2 3 4 5

Output : one

Input : 3

7 4 2

Output : zero

Question- 5

Question -: In a Super market we will find many variations of the same product. In the same way we can find many types of rice bags which differ in its quantity, price, brand, and type of rice in it. Many variations of same products exist in a super market. Consider rice for example. We get it in varying quantities (q) and at different prices (p).
Thus rice bag is unique combination of {q,p}. Customers want to buy the rice bags of their own choices. Each bag has two attributes price and the quantity of rice. The customers have some conditions for buying the rice bags, they have a specific price and quantity that have to be compared with the rice bags before buying them. If the price of rice bag is less than or equal to the customer's preference and if the quantity is more than given preference, then he/she will buy it. There is only one bag of each type and each customer can buy at most one bag.
Given n,m representing the number of customers and rice bags respectively, along with the variations of rice bags available in the market and the preferences of customers, find the maximum number of bags that can be sold by the end of the day.

Constraints
1 <= n, m <= 1000
1 <= a, b <= 10^5
1 <=p, q<= 10^5

Input
The first line contains two space separated integers n and m denoting the number of customers and number of rice bags respectively.
Next n lines consist of two space separated integers a and b denoting customer's preferences viz. customer's quantity and cost preferences, respectively.
Lastly, the next m lines consist of two space separated integers q and p denoting the bags quantity and cost, respectively.

Output
Print the maximum number of rice bags that can be sold.

Time Limit (secs)
1

Examples

Example 1
Input
5 4
35 2400
70 5500
87 6000
20 1700
12 500
50 2500
75 4875
100 7000
25 1250
Output
2
Explanation
Since price of bag should be less than or equal to customer's preference and the quantity should be greater than the preferred quantity, customer 2 can buy bag 2 and customer 4 can buy bag 4.
So, in total, 2 bags can be sold to the customers.

Example 2
4 7
32 1500
58 5000
87 6200
45 3000
20 1200
60 4500
100 6000
80 5500
35 1400
40 2000
50 2800
Output
4
Explanation
Since price of bag should be less than or equal to customer's preference and the quantity should be greater than the preferred quantity, customer 1 can buy bag 5, customer 2 can buy bag 2, customer 3 can buy bag 3 and customer 4 can buy 7th bag.
So, in total, 4 bags can be sold to the customers.

Question- 6

Harry Potter is hiking in the Alps surrounding Lake Geneva. In this area there are m cabins, numbered 1 to m. Each cabin is connected, with one or more trails, to a central meeting point next to the lake. Each trail is either short or long. Cabin i is connected with si short trails and li long trails to the lake.

Each day, Harry walks a trail from the cabin where he currently is to Lake Geneva, and then from there he walks a trail to any of the m cabins (including the one he started in). However, as he has to finish the hike in a day, at least one of the two trails has to be short.

How many possible combinations of trails can Harry take if he starts in cabin 1 and walks for n days?

Give the answer modulo 10^9+7.

Input
The first line contains the integers m and n.

The second line contains m integers, $s1,\ldots,sm$, where si is the number of short trails between cabin i and Lake Geneva.

The third and last line contains m integers, $l1,\ldots,lm$, where li is the number of long trails between cabin i and Lake Geneva.

We have the following constraints:

$0\leq si,li\leq 103$.

$1\leq m\leq 102$.

$1\leq n\leq 109$.

Output
The number of possible combinations of trails, modulo 10^9+7.

Example
input
3 2
1 0 1
0 1 1
output
18

Question- 7

A parentheses sequence is a string consisting of characters "(" and ")", for example "(()((".

A balanced parentheses sequence is a parentheses sequence which can become a valid mathematical expression after inserting numbers and operations into it, for example "(()(()))".

The balance of a parentheses sequence is defined as the number of opening parentheses "(" minus the number of closing parentheses ")". For example, the balance of the sequence "(()((" is 3.

A balanced parentheses sequence can also be defined as a parentheses sequence with balance 0 such that each of its prefixes has a non-negative balance.

We define the balanced shuffle operation that takes a parentheses sequence and returns a parentheses sequence as follows: first, for every character of the input sequence, we compute the balance of the prefix of the sequence before that character and write those down in a table together with the positions of the characters in the input sequence, for example:

Prefix balance	0	1	2	1	2	3	2	1
Position	1	2	3	4	5	6	7	8
Character	(	(	)	(	(	)	)	)

Then, we sort the columns of this table in increasing order of prefix balance, breaking ties in decreasing order of position. In the above example, we get:

Prefix balance	0	1	1	1	2	2	2	3
Position	1	8	4	2	7	5	3	6
Character	(	)	(	(	)	(	)	)

The last row of this table forms another parentheses sequence, in this case "()(()())". This sequence is called the result of applying the balanced shuffle operation to the input sequence, or in short just the balanced shuffle of the input sequence.

Surprisingly, it turns out that the balanced shuffle of any balanced parentheses sequence is always another balanced parentheses sequence (we will omit the proof for brevity). Even more surprisingly, the balanced shuffles of two different balanced parentheses sequences are always different, therefore the balanced shuffle operation is a bijection on the set of balanced parentheses sequences of any given length (we will omit this proof, too).

You are given a balanced parentheses sequence. Find its preimage: the balanced parentheses sequence the balanced shuffle of which is equal to the given sequence.

Input
The only line of input contains a string s consisting only of characters "(" and ")". This string is guaranteed to be a non-empty balanced parentheses sequence with its length not exceeding 500000.

Output
Print the balanced parentheses sequence t such that the balanced shuffle of t is equal to s. It is guaranteed that the answer always exists and is unique.

Example
input
()(()())
output
(()(()))

Question- 8

Like any unknown mathematician, Yuri has favourite numbers: A, B, C, and D, where $A \le B \le C \le D$. Yuri also likes triangles and once he thought: how many non-degenerate triangles with integer sides x, y, and z exist, such that $A \le x \le B \le y \le C \le z \le D$ holds?

Yuri is preparing problems for a new contest now, so he is very busy. That's why he asked you to calculate the number of triangles with described property.

The triangle is called non-degenerate if and only if its vertices are not collinear.

Input
The first line contains four integers: A, B, C and D ($1 \le A \le B \le C \le D \le 5 \cdot 105$) — Yuri's favourite numbers.

Output
Print the number of non-degenerate triangles with integer sides x, y, and z such that the inequality $A \le x \le B \le y \le C \le z \le D$ holds.

Examples
input
1 2 3 4
output
4

input
1 2 2 5
output
3

input
500000 500000 500000 500000
output
1

Note
In the first example Yuri can make up triangles with sides (1,3,3), (2,2,3), (2,3,3) and (2,3,4).

In the second example Yuri can make up triangles with sides (1,2,2), (2,2,2) and (2,2,3).

In the third example Yuri can make up only one equilateral triangle with sides equal to $5 \cdot 105$.

Question- 9

T is playing a game with his friend, HL.

There are n piles of stones, the i-th pile initially has ai stones.

T and HL will take alternating turns, with T going first. In each turn, a player chooses a non-empty pile and then removes a single stone from it. However, one cannot choose a pile that has been chosen in the previous turn (the pile that was chosen by the other player, or if the current turn is the first turn then the player can choose any non-empty pile). The player who cannot choose a pile in his turn loses, and the game ends.

Assuming both players play optimally, given the starting configuration of t games, determine the winner of each game.

Input
The first line of the input contains a single integer t ($1 \le t \le 100$) — the number of games. The description of the games follows. Each description contains two lines:

The first line contains a single integer n ($1 \le n \le 100$) — the number of piles.

The second line contains n integers $a1, a2, \ldots, an$ ($1 \le ai \le 100$).

Output
For each game, print on a single line the name of the winner, "T" or "HL" (without quotes)

Example
input
2
1
2
2
1 1
output
T
HL

Question- 10

An accordion is a string (yes, in the real world accordions are musical instruments, but let's forget about it for a while) which can be represented as a concatenation of: an opening bracket (ASCII code 091), a colon (ASCII code 058), some (possibly zero) vertical line characters (ASCII code 124), another colon, and a closing bracket (ASCII code 093). The length of the accordion is the number of characters in it.

For example, [::], [:||:] and [:|||:] are accordions having length 4, 6 and 7. (:|:), {:||:}, [:],]:||:[are not accordions.

You are given a string s. You want to transform it into an accordion by removing some (possibly zero) characters from it. Note that you may not insert new characters or reorder existing ones. Is it possible to obtain an accordion by removing characters from s, and if so, what is the maximum possible length of the result?

Input
The only line contains one string s ($1 \le |s| \le 500000$). It consists of lowercase Latin letters and characters [,], : and |.

Output
If it is not possible to obtain an accordion by removing some characters from s, print -1. Otherwise print maximum possible length of the resulting accordion.

Examples
inputCopy
|[a:b:|]
outputCopy
4
inputCopy
|]:[|:]
outputCopy
-1

Question- 11

A non-empty string is called palindrome, if it reads the same from the left to the right and from the right to the left. For example, "abcba", "a", and "abba" are palindromes, while "abab" and "xy" are not.

A string is called a substring of another string, if it can be obtained from that string by dropping some (possibly zero) number of characters from the beginning and from the end of it. For example, "abc", "ab", and "c" are substrings of the string "abc", while "ac" and "d" are not.

Let's define a palindromic count of the string as the number of its substrings that are palindromes. For example, the palindromic count of the string "aaa" is 6 because all its substrings are palindromes, and the palindromic count of the string "abc" is 3 because only its substrings of length 1 are palindromes.

You are given a string s. You can arbitrarily rearrange its characters. You goal is to obtain a string with the maximum possible value of palindromic count.

Input
The first line contains an integer n ($1 \le n \le 100000$) — the length of string s.

The second line contains string s that consists of exactly n lowercase characters of Latin alphabet.

Output
Print string t, which consists of the same set of characters (and each characters appears exactly the same number of times) as string s. Moreover, t should have the maximum possible value of palindromic count among all such strings strings.

If there are multiple such strings, print any of them.

Examples
input
5
oolol
output
ololo

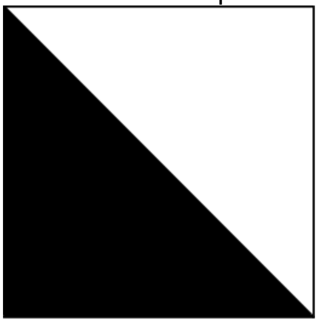
input
16
gagadbcgghhchbdf
output
abccbaghghghgdfd

Note
In the first example, string "ololo" has 9 palindromic substrings: "o", "l", "o", "l", "o", "olo", "lol", "olo", "ololo". Note, that even though some substrings coincide, they are counted as many times as they appear in the resulting string.

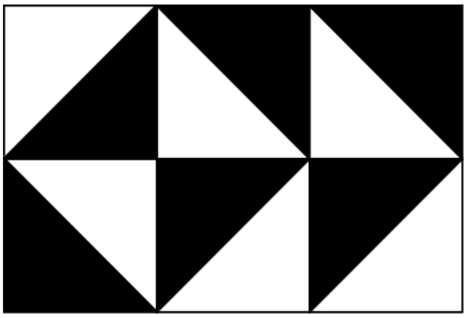
In the second example, the palindromic count of string "abccbaghghghgdfd" is 29.

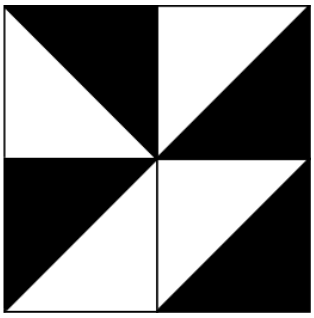
Question- 12

Bob is decorating his kitchen, more precisely, the floor. He has found a prime candidate for the tiles he will use. They come in a simple form factor — a square tile that is diagonally split into white and black part as depicted in the figure below.



The dimension of this tile is perfect for this kitchen, as he will need exactly $w \times h$ tiles without any scraps. That is, the width of the kitchen is w tiles, and the height is h tiles. As each tile can be rotated in one of four ways, he still needs to decide on how exactly he will tile the floor. There is a single aesthetic criterion that he wants to fulfil: two adjacent tiles must not share a colour on the edge — i.e. one of the tiles must have a white colour on the shared border, and the second one must be black.





The picture on the left shows one valid tiling of a 3×2 kitchen. The picture on the right shows an invalid arrangement, as the bottom two tiles touch with their white parts.
Find the number of possible tilings. As this number may be large, output its remainder when divided by 998244353 (a prime number).

Input
The only line contains two space separated integers w, h ($1 \leq w, h \leq 1000$) — the width and height of the kitchen, measured in tiles.

Output
Output a single integer n — the remainder of the number of tilings when divided by 998244353.

Examples
input
2 2
output
16

input
2 4
output
64

Question- 13

You are given a string s of length n consisting only of lowercase Latin letters.

A substring of a string is a contiguous subsequence of that string. So, string "forces" is substring of string "codeforces", but string "coder" is not.

Your task is to calculate the number of ways to remove exactly one substring from this string in such a way that all remaining characters are equal (the number of distinct characters either zero or one).

It is guaranteed that there is at least two different characters in s.

Note that you can remove the whole string and it is correct. Also note that you should remove at least one character.

Since the answer can be rather large (not very large though) print it modulo 998244353.

If you are Python programmer, consider using PyPy instead of Python when you submit your code.

Input
The first line of the input contains one integer n ($2 \le n \le (2x10^5)$) — the length of the string s.

The second line of the input contains the string s of length n consisting only of lowercase Latin letters.

It is guaranteed that there is at least two different characters in s.

Output
Print one integer — the number of ways modulo 998244353 to remove exactly one substring from s in such way that all remaining characters are equal.

Examples
input
4
abaa
output
6

input
7
aacdeee
output
6

input
2
az
output
3

Note
Let $s[l;r]$ be the substring of s from the position l to the position r inclusive.

Then in the first example you can remove the following substrings:

$s[1;2]$;
$s[1;3]$;
$s[1;4]$;
$s[2;2]$;
$s[2;3]$;
$s[2;4]$.

In the second example you can remove the following substrings:

$s[1;4]$;
$s[1;5]$;
$s[1;6]$;
$s[1;7]$;
$s[2;7]$;
$s[3;7]$.

In the third example you can remove the following substrings:

$s[1;1]$;
$s[1;2]$;
$s[2;2]$.

Question- 14

Today, Wet Shark is given n bishops on a 1000 by 1000 grid. Both rows and columns of the grid are numbered from 1 to 1000. Rows are numbered from top to bottom, while columns are numbered from left to right.

Wet Shark thinks that two bishops attack each other if they share the same diagonal. Note, that this is the only criteria, so two bishops may attack each other (according to Wet Shark) even if there is another bishop located between them. Now Wet Shark wants to count the number of pairs of bishops that attack each other.

Input
The first line of the input contains n ($1 \le n \le 200000$) — the number of bishops.

Each of next n lines contains two space separated integers xi and yi ($1 \le xi, yi \le 1000$) — the number of row and the number of column where i-th bishop is positioned. It's guaranteed that no two bishops share the same position.

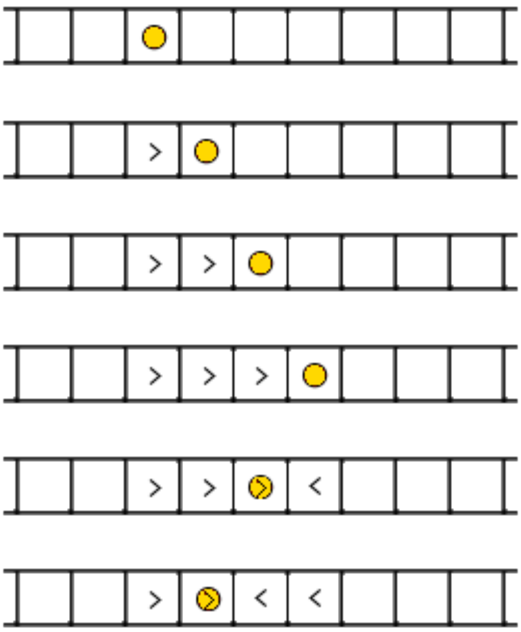
Output
Output one integer — the number of pairs of bishops which attack each other.

Examples
input
5
1 1
1 5
3 3
5 1
5 5
output
6
input
3
1 1
2 3
3 5
output
0

Note
In the first sample following pairs of bishops attack each other: (1,3), (1,5), (2,3), (2,4), (3,4) and (3,5). Pairs (1,2), (1,4), (2,5) and (4,5) do not attack each other because they do not share the same diagonal..

Question- 15

There is a straight snowy road, divided into n blocks. The blocks are numbered from 1 to n from left to right. If one moves from the i-th block to the (i+1)-th block, he will leave a right footprint on the i-th block. Similarly, if one moves from the i-th block to the (i-1)-th block, he will leave a left footprint on the i-th block. If there already is a footprint on the i-th block, the new footprint will cover the old one.



At the beginning, there were no footprints. Then polar bear Alice starts from the s-th block, makes a sequence of moves and ends in the t-th block. It is known that Alice never moves outside of the road.

You are given the description of Alice's footprints. Your task is to find a pair of possible values of s,t by looking at the footprints.

Input
The first line of the input contains integer n ($3 \leq n \leq 1000$).

The second line contains the description of the road — the string that consists of n characters. Each character will be either "." (a block without footprint), or "L" (a block with a left footprint), "R" (a block with a right footprint).

It's guaranteed that the given string contains at least one character not equal to ".". Also, the first and the last character will always be ".". It's guaranteed that a solution exists.

Output
Print two space-separated integers — the values of s and t. If there are several possible solutions you can print any of them.

Examples
input
9
..RRLL...
output
3 4

input
11
.RRRLLLLL..
output
7 5

Note
The first test sample is the one in the picture.