מטלה 1

1. Extra-long factional

The factorial of the integer n, written $n!$, is defined as:

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

Calculate and print the factorial of a given integer.

For example, if $n = 30$, we calculate $30 \times 29 \times 28 \times \cdots \times 2 \times 1$ and get $265252859812191058636308480000000$.

Function Description

Complete the extraLongFactorials function in the editor below. It should print the result and return.

extraLongFactorials has the following parameter(s):

- n: an integer

Note: Factorials of $n > 20$ can't be stored even in a $64 - bit$ long long variable. Big integers must be used for such calculations. Languages like Java, Python, Ruby etc. can handle big integers, but we need to write additional code in C/C++ to handle huge values.

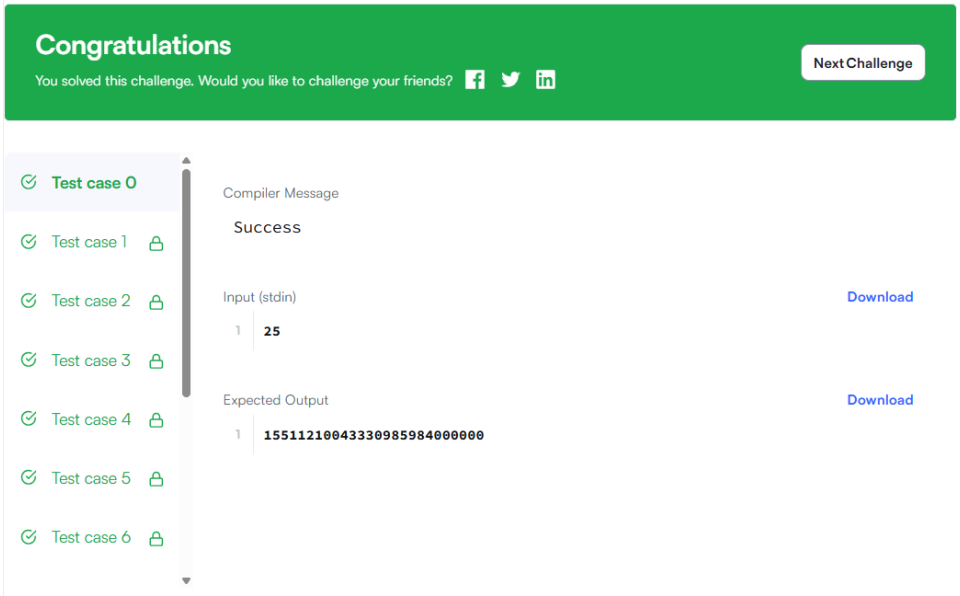
We recommend solving this challenge using BigIntegers.

Input Format

Input consists of a single integer n

Constraints

$1 \leq n \leq 100$



The time in words .2

Given the time in numerals we may convert it into words, as shown below:

$5:00 \rightarrow$ five o' clock
$5:01 \rightarrow$ one minute past five
$5:10 \rightarrow$ ten minutes past five
$5:15 \rightarrow$ quarter past five
$5:30 \rightarrow$ half past five
$5:40 \rightarrow$ twenty minutes to six
$5:45 \rightarrow$ quarter to six
$5:47 \rightarrow$ thirteen minutes to six
$5:28 \rightarrow$ twenty eight minutes past five

At $minutes = 0$, use o' clock. For $1 \leq minutes \leq 30$, use past, and for $30 < minutes$ use to. Note the space between the apostrophe and clock in o' clock. Write a program which prints the time in words for the input given in the format described.

Function Description

Complete the timeInWords function in the editor below.

timeInWords has the following parameter(s):

- int h: the hour of the day
- int m: the minutes after the hour

Returns

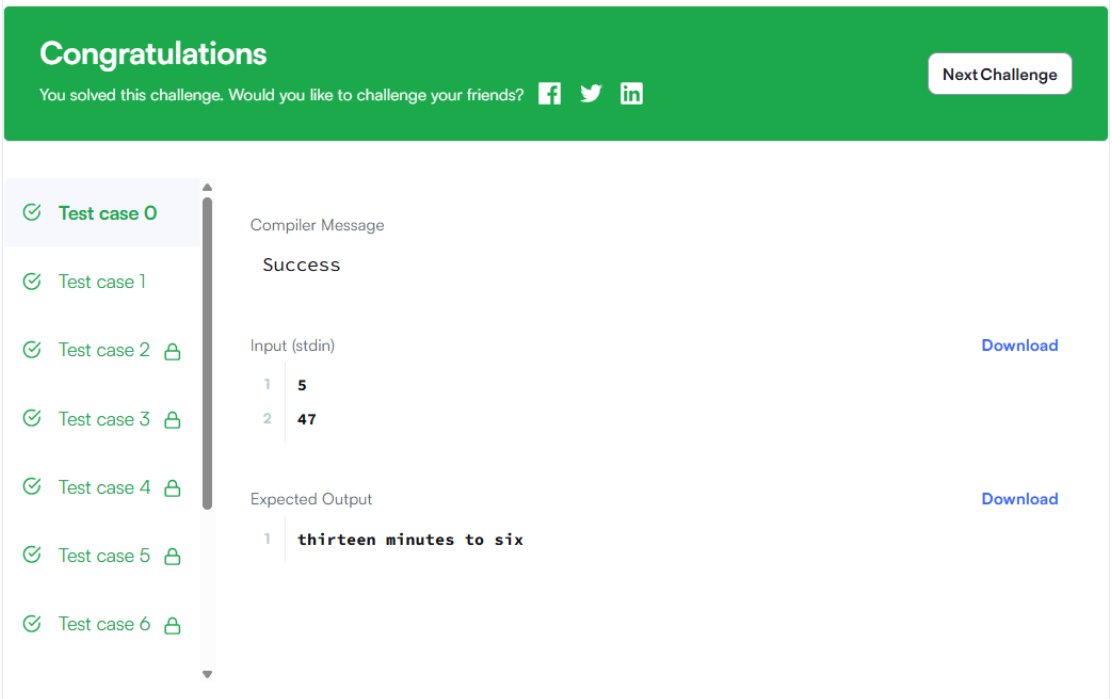
- string: a time string as described

Input Format

The first line contains h, the hours portion The second line contains m, the minutes portion

Constraints

- $1 \leq h \leq 12$
- $0 \leq m < 60$



Encryption .3

An English text needs to be encrypted using the following encryption scheme.

First, the spaces are removed from the text. Let L be the length of this text.

Then, characters are written into a grid, whose rows and columns have the following constraints:

$$\lfloor\sqrt{L}\rfloor \leq row \leq column \leq \lceil\sqrt{L}\rceil, \text{ where } \lfloor x\rfloor \text{ is floor function and } \lceil x\rceil \text{ is ceil function}$$

Example

$s =$ `if man was meant to stay on the ground god would have given us roots`

After removing spaces, the string is 54 characters long. $\sqrt{54}$ is between 7 and 8, so it is written in the form of a grid with 7 rows and 8 columns.

```
ifmanwas
meanttos
tayonthe
groundgo
dwouldha
vegivenu
sroots
```

- Ensure that $rows \times columns \geq L$
- If multiple grids satisfy the above conditions, choose the one with the minimum area, i.e. $rows \times columns$.

The encoded message is obtained by displaying the characters of each column, with a space between column texts. The encoded message for the grid above is:

`imtgdvs fearwer mayoogo anouuio ntnnlvt wttddes aohghn sseoau`

Create a function to encode a message.

Function Description

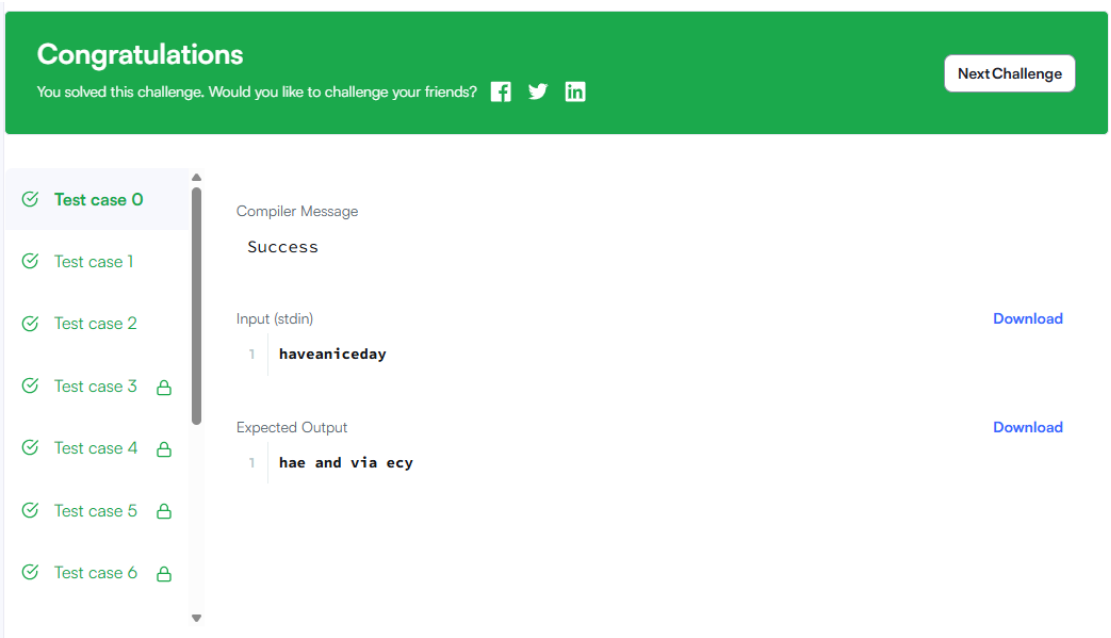
Complete the encryption function in the editor below.

encryption has the following parameter(s):

- string s: a string to encrypt

Returns

- string: the encrypted string



Madison is a little girl who is fond of toys. Her friend Mason works in a toy manufacturing factory . Mason has a 2D board A of size $H \times W$ with H rows and W columns. The board is divided into cells of size 1×1 with each cell indicated by its coordinate (i, j). The cell (i, j) has an integer A_{ij} written on it. To create the toy Mason stacks A_{ij} number of cubes of size $1 \times 1 \times 1$ on the cell (i, j).

Given the description of the board showing the values of A_{ij} and that the price of the toy is equal to the 3d surface area find the price of the toy.

Input Format

The first line contains two space-separated integers H and W the height and the width of the board respectively.

The next H lines contains W space separated integers. The j^{th} integer in i^{th} line denotes A_{ij}.

Constraints

- $1 \leq H, W \leq 100$
- $1 \leq A_{i,j} \leq 100$

Output Format

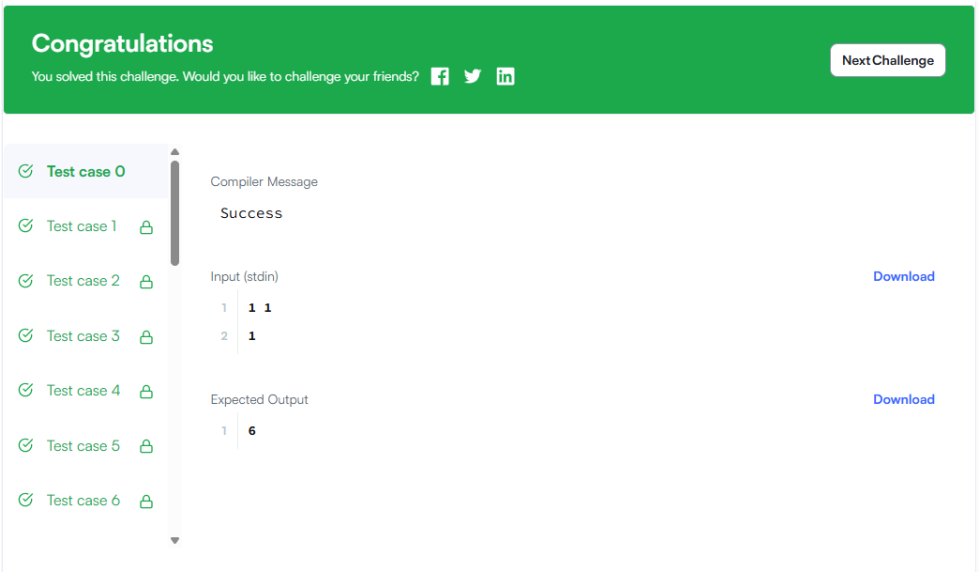
Print the required answer, i.e the price of the toy, in one line.

Sample Input 0

```
11
1
```

Sample Output 0

```
6
```



An arcade game player wants to climb to the top of the leaderboard and track their ranking. The game uses Dense Ranking, so its leaderboard works like this:

- The player with the highest score is ranked number 1 on the leaderboard.
- Players who have equal scores receive the same ranking number, and the next player(s) receive the immediately following ranking number.

Example

$ranked = [100, 90, 90, 80]$
$player = [70, 80, 105]$

The ranked players will have ranks 1, 2, 2, and 3, respectively. If the player's scores are 70, 80 and 105, their rankings after each game are 4^{th}, 3^{rd} and 1^{st}. Return $[4, 3, 1]$.

Function Description

Complete the climbingLeaderboard function in the editor below.

climbingLeaderboard has the following parameter(s):

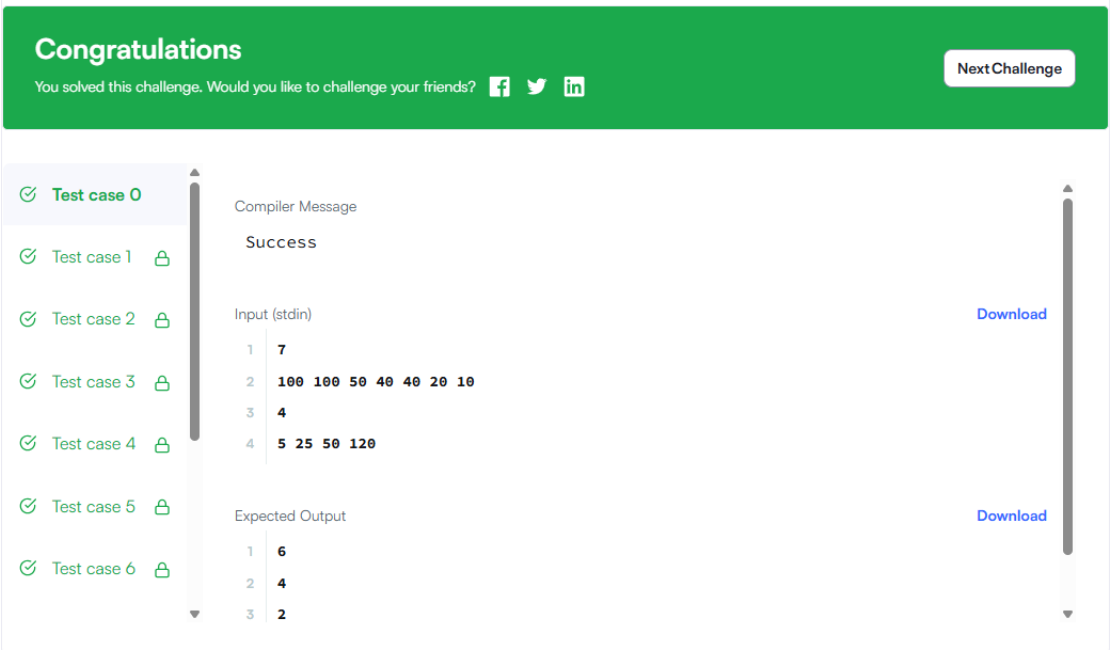
- int ranked[n]: the leaderboard scores
- int player[m]: the player's scores

Returns

- int[m]: the player's rank after each new score

Input Format

The first line contains an integer n, the number of players on the leaderboard.
The next line contains n space-separated integers $ranked[i]$, the leaderboard scores in decreasing order.
The next line contains an integer, m, the number games the player plays.
The last line contains m space-separated integers $player[j]$, the game scores.



Bigger Is Greater .6

Lexicographical order is often known as alphabetical order when dealing with strings. A string is greater than another string if it comes later in a lexicographically sorted list.

Given a word, create a new word by swapping some or all of its characters. This new word must meet two criteria:

- It must be greater than the original word
- It must be the smallest word that meets the first condition

Example

$w =$ `abcd`

The next largest word is `abdc`.

Complete the function biggerIsGreater below to create and return the new string meeting the criteria. If it is not possible, return `no answer`.

Function Description

Complete the biggerIsGreater function in the editor below.

biggerIsGreater has the following parameter(s):

- string w: a word

Returns

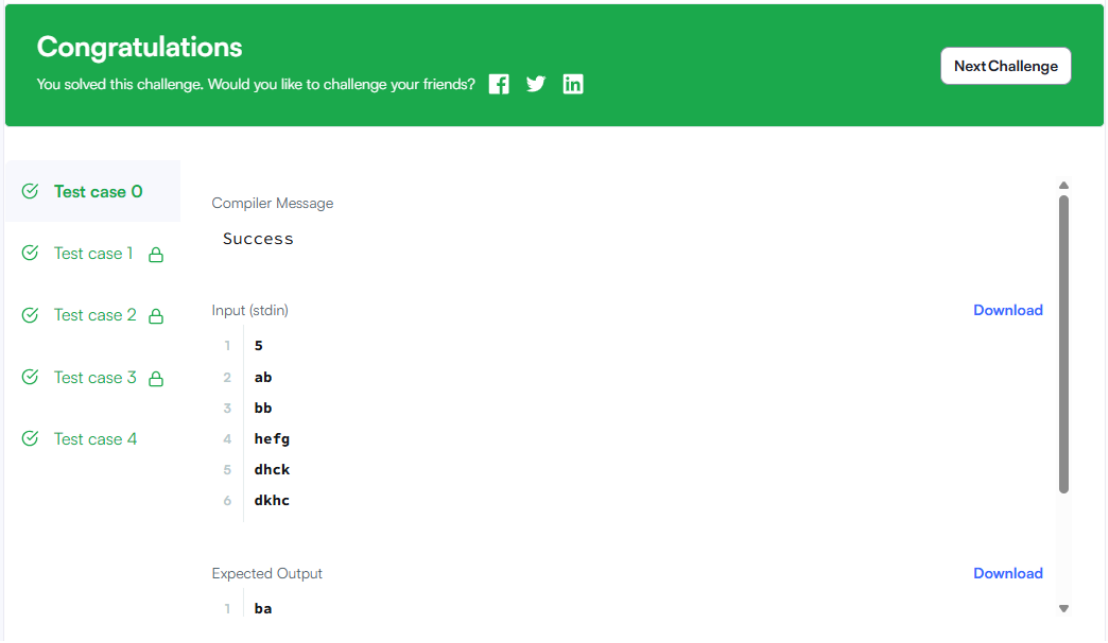
- string: the smallest lexicographically higher string possible or `no answer`

Input Format

The first line of input contains T, the number of test cases.
Each of the next T lines contains w.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq lengthofw \leq 100$
- w will contain only letters in the range ascii[a..z].



Given an array of strings of digits, try to find the occurrence of a given pattern of digits. In the grid and pattern arrays, each string represents a row in the grid. For example, consider the following grid:

```
1234567890
0987654321
1111111111
1111111111
2222222222
```

The pattern array is:

```
876543
111111
111111
```

The pattern begins at the second row and the third column of the grid and continues in the following two rows. The pattern is said to be present in the grid. The return value should be `YES` or `NO`, depending on whether the pattern is found. In this case, return `YES`.

Function Description

Complete the gridSearch function in the editor below. It should return `YES` if the pattern exists in the grid, or `NO` otherwise.

gridSearch has the following parameter(s):

- string G[R]: the grid to search
- string P[r]: the pattern to search for

Input Format

The first line contains an integer t, the number of test cases.

Each of the t test cases is represented as follows:

The first line contains two space-separated integers R and C, the number of rows in the search grid G and the length of each row string.

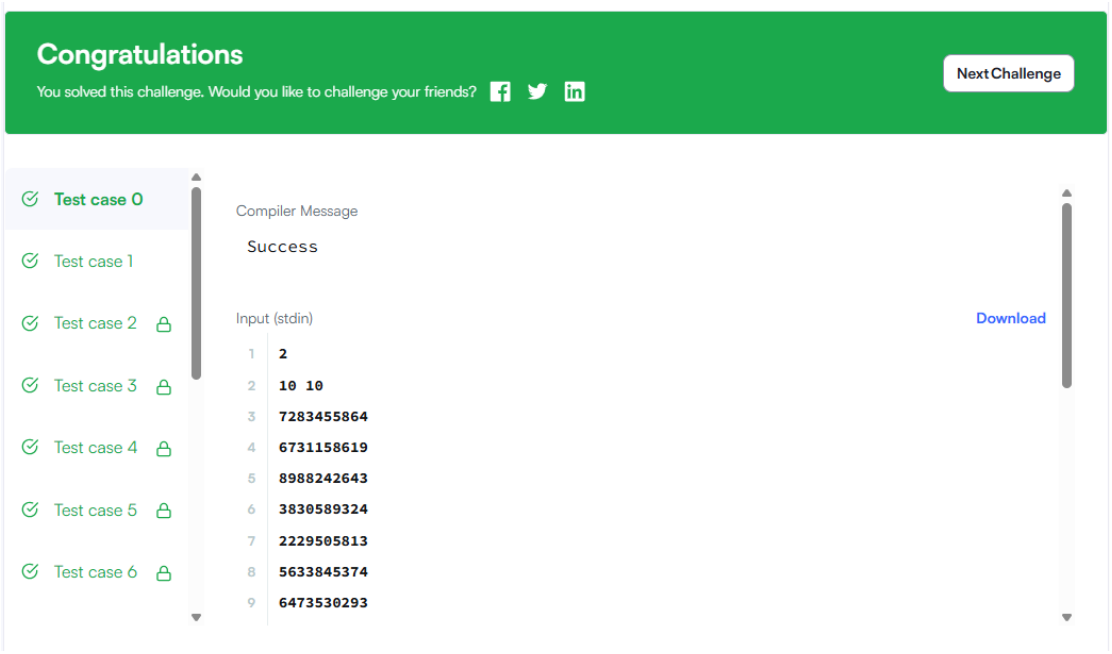
This is followed by R lines, each with a string of C digits that represent the grid G.

The following line contains two space-separated integers, r and c, the number of rows in the pattern grid P and the length of each pattern row string.

This is followed by r lines, each with a string of c digits that represent the pattern grid P.

Returns

- string: either `YES` or `NO`



Larry's Array .8

Larry has been given a permutation of a sequence of natural numbers incrementing from 1 as an array. He must determine whether the array can be sorted using the following operation any number of times:

- Choose any 3 consecutive indices and rotate their elements in such a way that $ABC \rightarrow BCA \rightarrow CAB \rightarrow ABC$.

For example, if $A = \{1, 6, 5, 2, 4, 3\}$:

```
A                rotate
[1,6,5,2,4,3]    [6,5,2]
[1,5,2,6,4,3]    [5,2,6]
[1,2,6,5,4,3]    [5,4,3]
[1,2,6,3,5,4]    [6,3,5]
[1,2,3,5,6,4]    [5,6,4]
[1,2,3,4,5,6]

YES
```

On a new line for each test case, print `YES` if A can be fully sorted. Otherwise, print `NO`.

Function Description

Complete the larrysArray function in the editor below. It must return a string, either `YES` or `NO`.

larrysArray has the following parameter(s):

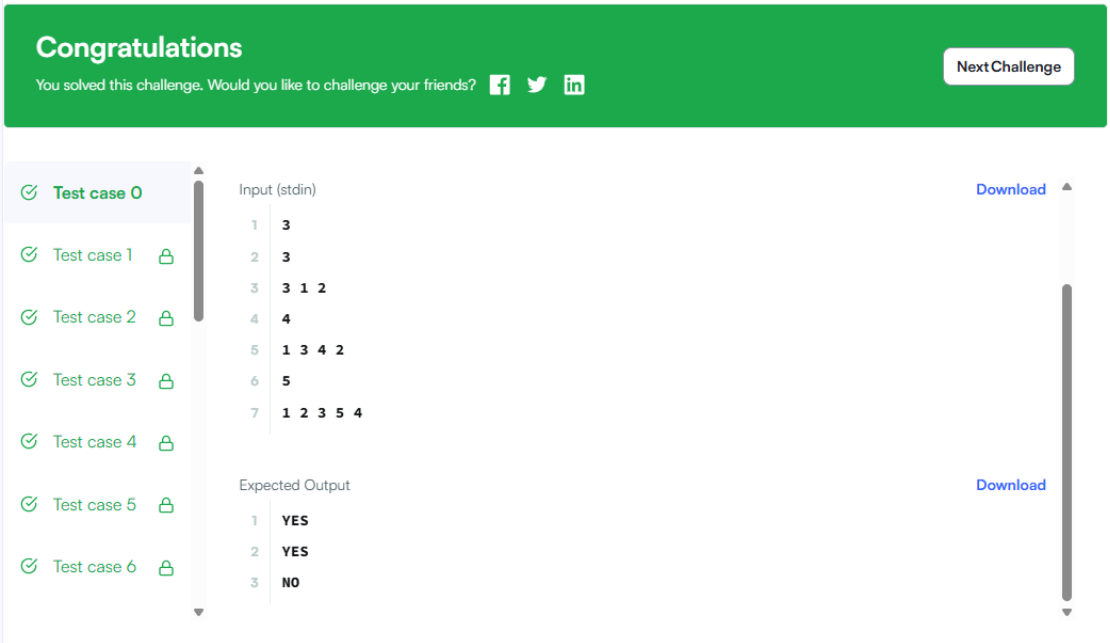
- A: an array of integers

Input Format

The first line contains an integer t, the number of test cases.

The next t pairs of lines are as follows:

- The first line contains an integer n, the length of A.
- The next line contains n space-separated integers $A[i]$.



Use the counting sort to order a list of strings associated with integers. If two strings are associated with the same integer, they must be printed in their original order, i.e. your sorting algorithm should be stable. There is one other twist: strings in the first half of the array are to be replaced with the character - (dash, ascii 45 decimal).

Insertion Sort and the simple version of Quicksort are stable, but the faster in-place version of Quicksort is not since it scrambles around elements while sorting.

Design your counting sort to be stable.

Example

$arr = [[0,'a'],[1,'b'],[0,'c'],[1,'d']]$

The first two strings are replaced with '-'. Since the maximum associated integer is 1, set up a helper array with at least two empty arrays as elements. The following shows the insertions into an array of three empty arrays.

i	string	converted list	
0			[[],[],[]]
1	a	-	[[-],[],[]]
2	b	-	[[-],[-],[]]
3	c		[[-,c],[-],[]]
4	d		[[-,c],[-,d],[]]

The result is then printed: - c - d .

Function Description

Complete the countSort function in the editor below. It should construct and print the sorted strings.

countSort has the following parameter(s):

- string arr[n][2]: each arr[i] is comprised of two strings, x and s

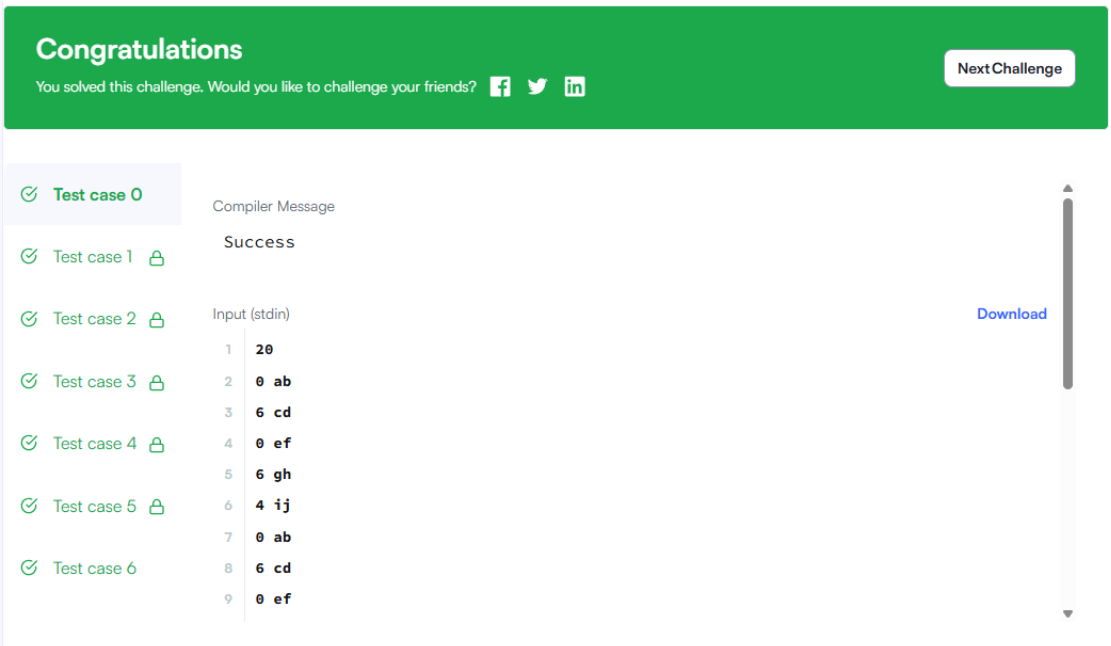
Returns

- Print the finished array with each element separated by a single space.

Note: The first element of each $arr[i]$, x, must be cast as an integer to perform the sort.

Input Format

The first line contains n, the number of integer/string pairs in the array arr.
Each of the next n contains $x[i]$ and $s[i]$, the integers (as strings) with their associated strings.



Sherlock And Anagram .10

Two strings are anagrams of each other if the letters of one string can be rearranged to form the other string. Given a string, find the number of pairs of substrings of the string that are anagrams of each other.

Example

$s = mom$

The list of all anagrammatic pairs is $[m, m]$, $[mo, om]$ at positions $[[0], [2]]$, $[[0, 1], [1, 2]]$ respectively.

Function Description

Complete the function sherlockAndAnagrams in the editor below.

sherlockAndAnagrams has the following parameter(s):

- string s: a string

Returns

- int: the number of unordered anagrammatic pairs of substrings in s

Input Format

The first line contains an integer q, the number of queries.

Each of the next q lines contains a string s to analyze.

Constraints

$1 \leq q \leq 10$

$2 \leq$ length of $s \leq 100$

s contains only lowercase letters in the range ascii[a-z].