question number 1

You are given an array of integers numbers and two integers left and right. You task is to calculate a boolean array result, where result[i] = true if there exists an integer x, such that numbers[i] = (i + 1) \* x and left ≤ x ≤ right. Otherwise, result[i] should be set to false.

Example

For numbers = [8, 5, 6, 16, 5], left = 1, and right = 3, the output should be solution(numbers, left, right) = [false, false, true, false, true].

For numbers[0] = 8, we need to find a value of x such that 1 \* x = 8, but the only value that would work is x = 8 which doesn't satisfy the boundaries 1 ≤ x ≤ 3, so result[0] = false.

For numbers[1] = 5, we need to find a value of x such that 2 \* x = 5, but there is no integer value that would satisfy this equation, so result[1] = false.

For numbers[2] = 6, we can choose x = 2 because 3 \* 2 = 6 and 1 ≤ 2 ≤ 3, so result[2] = true.

For numbers[3] = 16, there is no an integer 1 ≤ x ≤ 3, such that 4 \* x = 16, so result[3] = false.

For numbers[4] = 5, we can choose x = 1 because 5 \* 1 = 5 and 1 ≤ 1 ≤ 3, so result[4] = true.

Input/Output

[execution time limit] 4 seconds (dart)

[input] array.integer numbers

An array of integers.

Guaranteed constraints:

1 ≤ numbers.length ≤ 100,

1 ≤ numbers[i] ≤ 106.

[input] integer left

An integer representing the lower bound for x.

Guaranteed constraints:

1 ≤ left ≤ 104.

[input] integer right

An integer representing the upper bound for x.

Guaranteed constraints:

1 ≤ left ≤ right ≤ 104.

[output] array.boolean

A boolean array result described above.

[Dart] Syntax Tips

// Prints help message to the console

// Returns a string

String helloWorld(String name) {

print("This prints to the console when you Run Tests");

return "Hello, " + name;

}

test

Input:

numbers: [100]

left: 1

right: 1000

Expected Output:

[true]

Click the "Run Tests" button to see output and console logs.

test2

Input:

numbers: [1, 2, 3, 4, 5]

left: 1

right: 1

Expected Output:

[true, true, true, true, true]

Click the "Run Tests" button to see output and console logs.

test3

Input:

numbers: [1, 2, 3, 4, 5]

left: 2

right: 10000

Expected Output:

[false, false, false, false, false]

Click the "Run Tests" button to see output and console logs.

Input:

numbers: [1000000, 20000]

left: 10000

right: 10000

Expected Output:

[false, true]

Input:

numbers: [65, 46, 60, 164, 243, 228, 231, 298, 231, 211]

left: 20

right: 50

Expected Output:

[false, true, true, true, false, true, true, false, false, false]

Click the "Run Tests" button to see output and console logs.

question number 2:

Given two strings s and t, both consisting of lowercase English letters and digits, your task is to calculate how many ways exactly one digit could be removed from one of the strings so that s is lexicographically smaller than t after the removal. Note that we are removing only a single instance of a single digit, rather than all instances (eg: removing 1 from the string a11b1c could result in a1b1c or a11bc, but not abc).

Also note that digits are considered lexicographically smaller than letters.

Example

For s = "ab12c" and t = "1zz456", the output should be solution(s, t) = 1.

Here are all the possible removals:

We can remove the first digit from s, obtaining "ab2c". "ab2c" > "1zz456", so we don't count this removal

We can remove the second digit from s, obtaining "ab1c". "ab1c" > "1zz456", so we don't count this removal

We can remove the first digit from t, obtaining "zz456". "ab12c" < "zz456", so we count this removal

We can remove the second digit from t, obtaining "1zz56". "ab12c" > "1zz56", so we don't count this removal

We can remove the third digit from t, obtaining "1zz46". "ab12c" > "1zz46", so we don't count this removal

We can remove the fourth digit from t, obtaining "1zz45". "ab12c" > "1zz45", so we don't count this removal

The only valid case where s < t after removing a digit is "ab12c" < "zz456". Therefore, the answer is 1.

For s = "ab12c" and t = "ab24z", the output should be solution(s, t) = 3.

There are 4 possible ways of removing the digit:

"ab1c" < "ab24z"

"ab2c" > "ab24z"

"ab12c" < "ab4z"

"ab12c" < "ab2z"

Three of these cases match the requirement that s < t, so the answer is 3.

Input/Output

[execution time limit] 4 seconds (dart)

[input] string s

A string consisting of lowercase English letters and digits 0..9.

Guaranteed constraints:

1 ≤ s.length ≤ 103.

[input] string t

A string consisting of lowercase English letters and digits 0..9.

Guaranteed constraints:

1 ≤ t.length ≤ 103.

[output] integer

The number of ways to remove exactly one digit from one of the strings so that s is lexicographically smaller than t after the removal.

[Dart] Syntax Tips

// Prints help message to the console

// Returns a string

String helloWorld(String name) {

print("This prints to the console when you Run Tests");

return "Hello, " + name;

}

tests

Input:

s: "ab12c"

t: "1zz456"

Expected Output:

1

Click the "Run Tests" button to see output and console logs.

Input:

s: "ab12c"

t: "ab24z"

Expected Output:

3

Click the "Run Tests" button to see output and console logs.

Test 3

Input:

s: "96726"

t: "9z34c"

Expected Output:

8

Click the "Run Tests" button to see output and console logs.

Input:

s: "4u05q"

t: "ed0r7"

Expected Output:

4

Click the "Run Tests" button to see output and console logs.

Input:

s: "6"

t: "h"

Expected Output:

1

Click the "Run Tests" button to see output and console logs.

Input:

s: "5"

t: "4"

Expected Output:

1

Click the "Run Tests" button to see output and console logs.

Input:

s: "d"

t: "q"

Expected Output:

0

Click the "Run Tests" button to see output and console logs.

questions number 3

You are given an array of arrays a. Your task is to group the arrays a[i] by their mean values, so that arrays with equal mean values are in the same group, and arrays with different mean values are in different groups.

Each group should contain a set of indices (i, j, etc), such that the corresponding arrays (a[i], a[j], etc) all have the same mean. Return the set of groups as an array of arrays, where the indices within each group are sorted in ascending order, and the groups are sorted in ascending order of their minimum element.

Example

For

a = [[3, 3, 4, 2],

[4, 4],

[4, 0, 3, 3],

[2, 3],

[3, 3, 3]]

the output should be

solution(a) = [[0, 4],

[1],

[2, 3]]

mean(a[0]) = (3 + 3 + 4 + 2) / 4 = 3;

mean(a[1]) = (4 + 4) / 2 = 4;

mean(a[2]) = (4 + 0 + 3 + 3) / 4 = 2.5;

mean(a[3]) = (2 + 3) / 2 = 2.5;

mean(a[4]) = (3 + 3 + 3) / 3 = 3.

There are three groups of means: those with mean 2.5, 3, and 4. And they form the following groups:

Arrays with indices 0 and 4 form a group with mean 3;

Array with index 1 forms a group with mean 4;

Arrays with indices 2 and 3 form a group with mean 2.5.

Note that neither

solution(a) = [[0, 4],

[2, 3],

[1]]

nor

solution(a) = [[0, 4],

[1],

[3, 2]]

will be considered as a correct answer:

In the first case, the minimal element in the array at index 2 is 1, and it is less then the minimal element in the array at index 1, which is 2.

In the second case, the array at index 2 is not sorted in ascending order.

For

a = [[-5, 2, 3],

[0, 0],

[0],

[-100, 100]]

the output should be

solution(a) = [[0, 1, 2, 3]]

The mean values of all of the arrays are 0, so all of them are in the same group.

Input/Output

[execution time limit] 4 seconds (dart)

[input] array.array.integer a

An array of arrays of integers.

Guaranteed constraints:

1 ≤ a.length ≤ 100,

1 ≤ a[i].length ≤ 100,

-100 ≤ a[i][j] ≤ 100.

[output] array.array.integer

An array of arrays, representing the groups of indices.

[Dart] Syntax Tips

// Prints help message to the console

// Returns a string

String helloWorld(String name) {

print("This prints to the console when you Run Tests");

return "Hello, " + name;

}

test

Input:

a:

[[3,3,4,2],

[4,4],

[4,0,3,3],

[2,3],

[3,3,3]]

Expected Output:

[[0,4],

[1],

[2,3]]

Click the "Run Tests" button to see output and console logs.

Input:

a:

[[-5,2,3],

[0,0],

[0],

[-100,100]]

Expected Output:

[[0,1,2,3]]

Click the "Run Tests" button to see output and console logs.

Test 4

Input:

a:

[[2,2,-3],

[1],

[-10],

[7]]

Expected Output:

[[0],

[1],

[2],

[3]]

Click the "Run Tests" button to see output and console logs.

a:

[[0],

[0,0],

[0,1],

[1,1,0,1,1],

[1,0],

[1],

[0],

[0,0,1,0,1,0],

[1,0,0,1],

[1,1,1],

[0,0,0],

[1,1],

[0,1,0,0,1]]

Expected Output:

[[0,1,6,10],

[2,4,8],

[3],

[5,9,11],

[7],

[12]]

Click the "Run Tests" button to see output and console logs.

question number 4

You've created a new programming language, and now you've decided to add hashmap support to it. Actually you are quite disappointed that in common programming languages it's impossible to add a number to all hashmap keys, or all its values. So you've decided to take matters into your own hands and implement your own hashmap in your new language that has the following operations:

insert x y - insert an object with key x and value y.

get x - return the value of an object with key x.

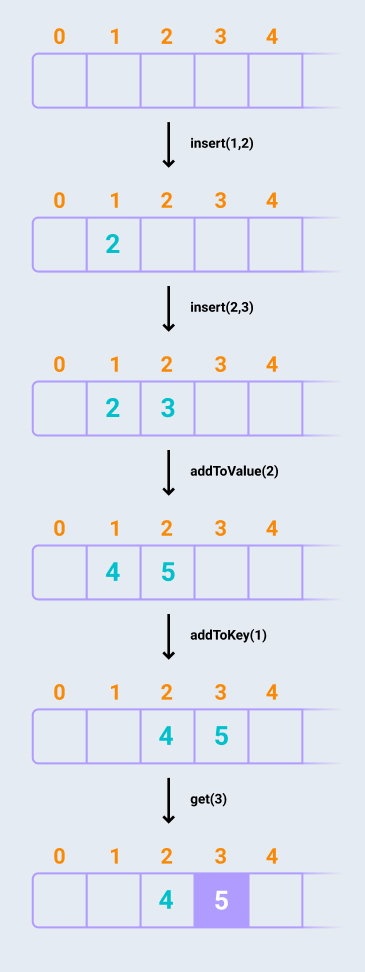
addToKey x - add x to all keys in map.

addToValue y - add y to all values in map.

To test out your new hashmap, you have a list of queries in the form of two arrays: queryTypes contains the names of the methods to be called (eg: insert, get, etc), and queries contains the arguments for those methods (the x and y values).

Your task is to implement this hashmap, apply the given queries, and to find the sum of all the results for get operations.

Example



For queryType = ["insert", "insert", "addToValue", "addToKey", "get"] and query = [[1, 2], [2, 3], [2], [1], [3]], the output should be solution(queryType, query) = 5.

The hashmap looks like this after each query:

1 query: {1: 2}

2 query: {1: 2, 2: 3}

3 query: {1: 4, 2: 5}

4 query: {2: 4, 3: 5}

5 query: answer is 5

The result of the last get query for 3 is 5 in the resulting hashmap.

For queryType = ["insert", "addToValue", "get", "insert", "addToKey", "addToValue", "get"] and query = [[1, 2], [2], [1], [2, 3], [1], [-1], [3]], the output should be solution(queryType, query) = 6.

The hashmap looks like this after each query:

1 query: {1: 2}

2 query: {1: 4}

3 query: answer is 4

4 query: {1: 4, 2: 3}

5 query: {2: 4, 3: 3}

6 query: {2: 3, 3: 2}

7 query: answer is 2

The sum of the results for all the get queries is equal to 4 + 2 = 6.

Input/Output

[execution time limit] 4 seconds (dart)

[input] array.string queryType

Array of query types. It is guaranteed that each queryType[i] is either "addToKey", "addToValue", "get", or "insert".

Guaranteed constraints:

1 ≤ queryType.length ≤ 105.

[input] array.array.integer query

Array of queries, where each query is represented either by two numbers for insert query or by one number for other queries. It is guaranteed that during all queries all keys and values are in the range [-109, 109].

Guaranteed constraints:

query.length = queryType.length,

1 ≤ query[i].length ≤ 2.

[output] integer64

The sum of the results for all get queries.

[Dart] Syntax Tips

// Prints help message to the console

// Returns a string

String helloWorld(String name) {

print("This prints to the console when you Run Tests");

return "Hello, " + name;

}

Test 1

Input:

queryType:

["insert",

"insert",

"addToValue",

"addToKey",

"get"]

query:

[[1,2],

[2,3],

[2],

[1],

[3]]

Expected Output:

5

Click the "Run Tests" button to see output and console logs.

Test 2

Input:

queryType:

["insert",

"addToValue",

"get",

"insert",

"addToKey",

"addToValue",

"get"]

query:

[[1,2],

[2],

[1],

[2,3],

[1],

[-1],

[3]]

Expected Output:

6

Click the "Run Tests" button to see output and console logs.

queryType:

["addToKey",

"addToKey",

"insert",

"addToValue",

"addToValue",

"get",

"addToKey",

"insert",

"addToKey",

"addToValue"]

query:

[[-3],

[-1],

[0,-3],

[3],

[-1],

[0],

[-1],

[-4,-5],

[-1],

[-4]]

Expected Output:

-1

Click the "Run Tests" button to see output and console logs.