Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/K36biX4EgomKmiYTj

Gmail | YouTube | Google | LinkedIn | Spotify – Web Player | WhatsApp | ChatGPT | RESOURCE HUB | Striver's SDE Sheet... | Strivers A2Z DSA Co... | Problems - LeetCode | DeepSeek - Into the...

BACK TO QUESTIONS

Question 1 of 4

1:59:25

DESC

HISTORY

RULES

INFO

SETTINGS

You're creating a new programming language with some exciting new features! Any programming language can check if two strings are matching, but you'd like yours to be able to check if they're *almost matching*. More specifically, we'll say two strings are *almost matching* if they're equal in length and all of their corresponding characters are the same except for one. For example, `"cat"` and `"bat"` are *almost matching*, but `"cat"` and `"dog"` are not.

For the sake of efficiency, you're planning on testing the feature by using a single string and comparing its substrings. Given a string `s` and an integer `k`, your task is to find the number of pairs of substrings of `s` that are *almost matching* but differ at their `k`th character (`0`-based). It's necessary that the length of both substrings exceeds `k` (otherwise the strings wouldn't have a `k`th character).

Also note that substrings are determined by their indices, so there could potentially be multiple instances of the same word. For example, in the word `"ingratiating"` the substring `"ing"` beginning at index `0` is considered distinct from the

main.cpp

C++

```
int solution(string s, int k) {

}
```

00:04:31

01:25:28

TESTS

CUSTOM TESTS

RUN TESTS

MORE

VIEW DIFF

SUBMIT

NEXT

20250811_192820

28°C Cloudy

Search

ENG IN

01:58 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/K36biX4EgomKmiYTj

BACK TO QUESTIONS

Question 1 of 4

1:55:56

DESC

HISTORY

RULES

INFO

SETTINGS

Example

For `s = "abacaba"` and `k = 1`, the output should be `solution(s, k) = 8`.

Using `i` and `j` to represent the start and end indices of the first substrings, and `l` and `m` to represent the indices of the second substring, the `8` pairs are:

- `("aba", "aca")` - `i = 0, j = 2, l = 2, m = 4`;
- `("aba", "aca")` - `i = 4, j = 6, l = 2, m = 4`;
- `("aca", "aba")` - `i = 2, j = 4, l = 0, m = 2`;
- `("aca", "aba")` - `i = 2, j = 4, l = 4, m = 6`;
- `("ac", "ab")` - `i = 2, j = 3, l = 0, m = 1`;
- `("ac", "ab")` - `i = 2, j = 3, l = 4, m = 5`;

main.cpp

C++

```
int solution(string s, int k) {
    int n=s.size();
}
```

00:08:01

01:21:58

MORE

VIEW DIFF

SUBMIT

NEXT

20250811_192820

28°C Cloudy

Search

ENG IN

01:59 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/Q2QMZCYchfjQpkn6N

BACK TO QUESTIONS

Question 2 of 4

1:30:00

DESC

HISTORY

RULES

INFO

SETTINGS

The second argument is an integer B

The third argument is an integer array C

Output Format

Return an integer array representing the answer.

Example Input

Input 1 :

A = 7

B = 2

C = [1, 2, 1]

Input 2 :

A = 10

B = 5

C = [2]

Example Output

Output1 :

[0, 1, 0, 2, 2, 0, 3]

main.cpp Saved

C++

```
vector<int> solution(int a, int b, vector<int> c) {

}
```

00:33:56

00:56:03

MORE

VIEW DIFF

SUBMIT

NEXT

20250811_192820

28°C Cloudy

Search

ENG IN

02:01 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/Q2QMZCYchfjQpkn6N

Gmail | YouTube | Google | LinkedIn | Spotify – Web Player | WhatsApp | ChatGPT | RESOURCE HUB | Striver's SDE Sheet... | Strivers A2Z DSA Co... | Problems - LeetCode | DeepSeek - Into the...

BACK TO QUESTIONS | Question 2 of 4 | 1:30:04

DESC | HISTORY | RULES | INFO | SETTINGS

C = [2]

Example Output

Output1 :

[0, 1, 0, 2, 2, 0, 3]

Output2 :

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

Example Explanation

For Input 1:

The lexicographically smallest possible answer is [0, 1, 0, 2, 2, 0, 3].
The sequence of jumps Mario performs is 0 → 2 → 4 → 5 → 7 → 8.

For Input2:

The lexicographically smallest possible answer is [0, 0, 0, 0, 1, 1, 0, 0, 0, 0].
The sequence of jumps you perform is 0 → 5 → 6 → 11

main.cpp | Saved | C++

```
vector<int> solution(int a, int b, vector<int> c) {

}
```

00:33:53 | 00:56:06

MORE | VIEW DIFF | SUBMIT | NEXT

20250811_192820

28°C Cloudy | Search | ENG IN | 02:01 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/Q2QMZCYchfjQpkn6N

BACK TO QUESTIONS

Question 2 of 4

1:31:30

DESC

HISTORY

RULES

INFO

SETTINGS

Problem Description

Welcome to the magical world of Mario's adventure where Mario is standing on platform 0 and needs to reach platform A+1. But oh no, there's a wide river that's A units wide blocking his way. Mario can't walk on water, but he's got an amazing ability - he can jump up to B units at a time, either forward or backward.

On the river, there are various-sized wooden platforms floating on the river just like in Mario's games. In total there are M platforms and their sizes are stored in an integer array C. Mario can move these platforms around, but he can't change their order or let them overlap. Note that he should first move the platforms and then start his journey. Help Mario cross the river and reach platform A+1.

If it is impossible to reach A+1, your answer will be a vector containing -1, acknowledging the challenge. Yet, if a solution exists, return a vector of size A representing the sequence of river cells. If cell i does not belong to any platform, the ith element of

main.cpp Saved

C++

```
vector<int> solution(int a, int b, vector<int> c) {

}
```

00:32:27 00:57:32

MORE

VIEW DIFF SUBMIT NEXT

20250811_192820

28°C Cloudy

Search

ENG IN

02:00 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/Q2QMZCYchfjQpkn6N

BACK TO QUESTIONS

Question 2 of 4

1:30:15

DESC

HISTORY

RULES

INFO

SETTINGS

the lexicographically smallest one.

Problem Constraints

1 <= A <= 10^5

1 <= B <= 10^5

1 <= M <= A

1<= C[i] <= A

ΣC[i] <= A

Input Format

The first argument is an integer A

The second argument is an integer B

The third argument is an integer array C

Output Format

Return an integer array representing the answer.

Example Input

Input 1 :

main.cpp Saved

C++

```
vector<int> solution(int a, int b, vector<int> c) {

}
```

TESTS CUSTOM TESTS

RUN TESTS

00:33:22

00:56:37

MORE

VIEW DIFF SUBMIT NEXT

20250811_192820

28°C Cloudy

Search

ENG IN

02:00 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/iNAydD4mKMZbiHWnb

BACK TO QUESTIONS

Question 3 of 4

0:47:39

DESC

HISTORY

RULES

INFO

SETTINGS

```
...#..
.##...
```

Your task is to calculate the minimum and the maximum number of moves required to fill the first column of the board with blocks.

Example

- For

```
field = [['.', '#', '#'],
         ['#', '.', '.'],
         ['.', '.', '.']]
```

the output should be

`solution(field) = [4, 4]`.

We need to choose the first row once and the third row three times.

- For

main.cpp

C++

```cpp
vector<int> solution(vector<vector<char>> field) {

}
```

01:16:19

00:13:40

MORE

VIEW DIFF

SUBMIT

NEXT

20250811_192820

Gold -1.35%

Search

ENG IN

02:04 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/iNAydD4mKMZbiHWnb

BACK TO QUESTIONS

Question 3 of 4

0:48:13

DESC

HISTORY

RULES

INFO

SETTINGS

```
...#..
.#....
```

blocks are denoted by `'#'` and empty cells are denoted by `'.'`

and you throw a block into the first or the second row from the top, it will end up like this:

```
...#..
...#..
.##...
```

Your task is to calculate the minimum and the maximum number of moves required to fill the first column of the board with blocks.

Example

- For

```
field = [['.', '#', '#'],
         ['#', '.', '.'],
         ['.', '.', '.']]
```

the output should be

main.cpp

C++

```
vector<int> solution(vector<vector<char>> field) {

}
```

01:15:44

00:14:15

MORE

VIEW DIFF

SUBMIT

NEXT

20250811_192820

Gold -1.35%

Search

ENG IN

02:04 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/iNAydD4mKMZbiHWnb

BACK TO QUESTIONS

Question 3 of 4

0:49:15

DESC

HISTORY

RULES

INFO

SETTINGS

You are given a rectangular board divided into a uniform grid (square cells). Some cells of the board are occupied with blocks, and others are empty. You are trying to add more and more blocks to the board, and your task is to fill the first column with them. Your can add a block to the field in the following way: first, you choose the row index, then you throw the new block into the chosen row from the left. The block appears in the leftmost cell of the row and starts moving to the right, until it reaches another block or the end of the row. When that happens, the block starts falling down until in reaches another block or the last row.

For example, if the board looks like this

```
...#..
...#..
.#....
```

blocks are denoted by `'#'` and empty cells are denoted by `'.'`

and you throw a block into the first or the second

main.cpp

C++

```cpp
vector<int> solution(vector<vector<char>> field) {

}
```

01:14:41

00:15:18

MORE

VIEW DIFF

SUBMIT

NEXT

20250811_192820

Gold -1.35%

Search

ENG IN

02:03 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/iNAydD4mKMZbiHWnb

Gmail | YouTube | Google | LinkedIn | Spotify – Web Player | WhatsApp | ChatGPT | RESOURCE HUB | Striver's SDE Sheet... | Strivers A2Z DSA Co... | Problems - LeetCode | DeepSeek - Into the...

BACK TO QUESTIONS

Question 3 of 4

0:48:21

DESC | HISTORY | RULES | INFO | SETTINGS

block appears in the leftmost cell of the row and starts moving to the right, until it reaches another block or the end of the row. When that happens, the block starts falling down until in reaches another block or the last row.

For example, if the board looks like this

```
...#..
...#..
.#....
```

blocks are denoted by `'#'` and empty cells are denoted by `'.'`

and you throw a block into the first or the second row from the top, it will end up like this:

```
...#..
...#..
.##...
```

Your task is to calculate the minimum and the maximum number of moves required to fill the first column of the board with blocks.

main.cpp

C++

```cpp
vector<int> solution(vector<vector<char>> field) {

}
```

01:15:36 00:14:23

MORE | VIEW DIFF | SUBMIT | NEXT

20250811_192820

Gold -1.35%

Search

ENG IN 02:03 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/wikk4EXjs2KKmaLEu

Gmail | YouTube | Google | LinkedIn | Spotify – Web Player | WhatsApp | ChatGPT | RESOURCE HUB | Striver's SDE Sheet... | Strivers A2Z DSA Co... | Problems - LeetCode | DeepSeek - Into the...

BACK TO QUESTIONS

Question 4 of 4

1:24:20

DESC | HISTORY | RULES | INFO | SETTINGS

A *cryptarithm* is a mathematical puzzle where the goal is to find the correspondence between letters and digits such that the given arithmetic equation consisting of letters holds true.

Given a *cryptarithm* as an array of strings `crypt`, count the number of its valid solutions.

The solution is valid if each letter represents a different digit, and the leading digit of any multi-digit number is not zero.

`crypt` has the following structure: `[word1, word2, word3]`, which stands for the `word1 + word2 = word3` *cryptarithm*.

Example

- For `crypt = ["SEND", "MORE", "MONEY"]`, the output should be `solution(crypt) = 1`, because there is only one solution to this *cryptarithm*: `O = 0`, `M = 1`, `Y = 2`, `E = 5`, `N = 6`, `D = 7`, `R = 8`, and `S = 9` (9567 + 1085 = 10652).

main.cpp

C++

```
int solution(vector<string> crypt) {

}
```

00:39:37 00:50:22

MORE | VIEW DIFF | SUBMIT | REVIEW YOUR ANSWERS

20250811_192820

28°C Cloudy | Search | ENG IN | 02:02 12-08-2025

Media Player

https://app.codesignal.com/test/yRivuLQPjgbXzZAQo/question/wikk4EXjs2KKmaLEu

BACK TO QUESTIONS

Question 4 of 4

1:23:31

DESC

HISTORY

RULES

INFO

SETTINGS

Example

- For `crypt = ["SEND", "MORE", "MONEY"]`, the output should be `solution(crypt) = 1`, because there is only one solution to this *cryptarithm*: `O = 0`, `M = 1`, `Y = 2`, `E = 5`, `N = 6`, `D = 7`, `R = 8`, and `S = 9` (`9567 + 1085 = 10652`).
- For `crypt = ["GREEN", "BLUE", "BLACK"]`, the output should be `solution(crypt) = 12`, because there are `12` possible valid solutions:
  - `54889 + 6134 = 61027`
  - `18559 + 2075 = 20634`
  - `72449 + 8064 = 80513`
  - `48229 + 5372 = 53601`
  - `47119 + 5261 = 52380`
  - `36887 + 4028 = 40915`
  - `83447 + 9204 = 92651`
  - `74665 + 8236 = 82901`
  - `65884 + 7308 = 73192`
  - `57883 + 6408 = 64291`
  - `57881 + 6428 = 64309`

main.cpp

C++

```
int solution(vector<string> crypt) {

}
```

00:40:26

00:49:33

MORE

VIEW DIFF

SUBMIT

REVIEW YOUR ANSWERS

20250811_192820

Gold -1.35%

Search

ENG IN

02:02 12-08-2025