HackerRank

Hey, Alejandro

Help us personalize your experience by telling us a bit about yourself.

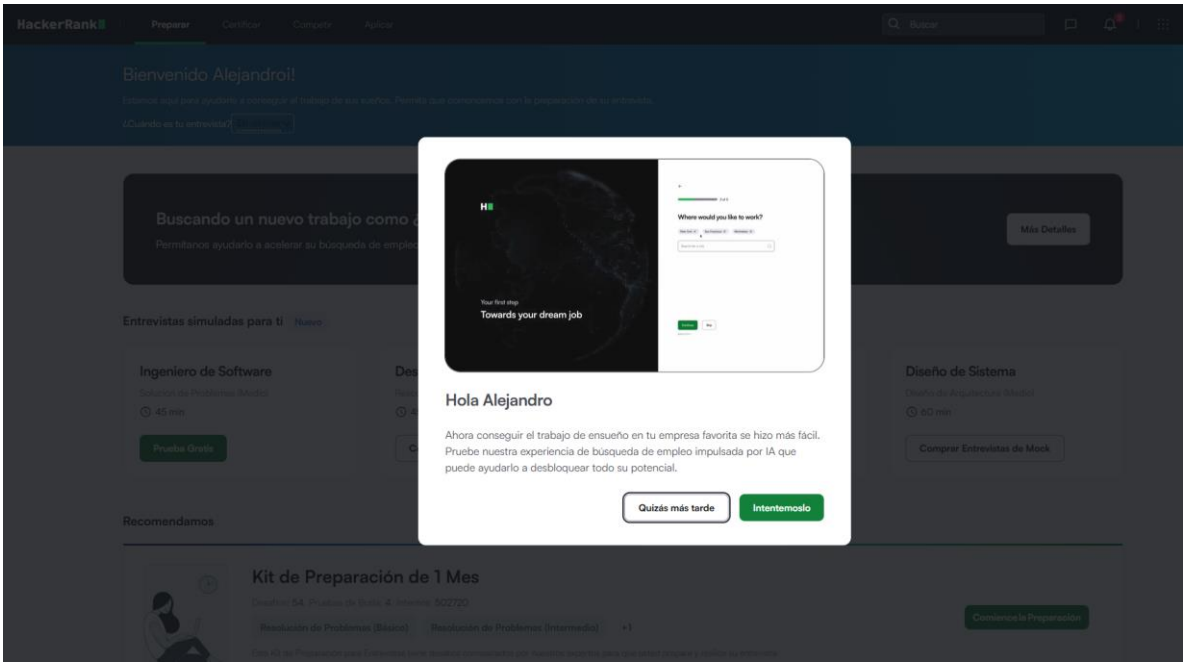
Terms Of Service | Privacy Policy

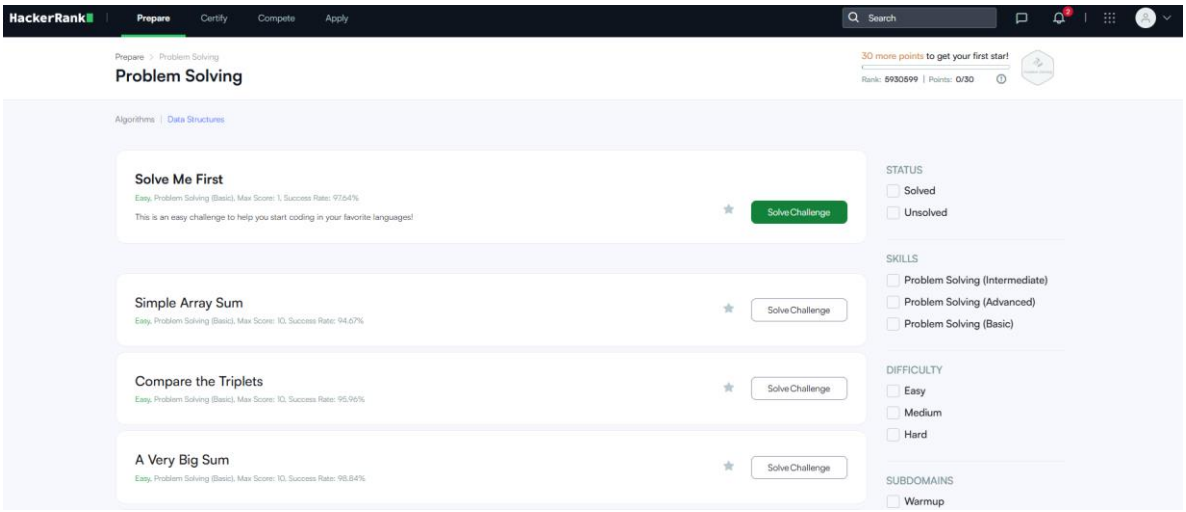
Step 1

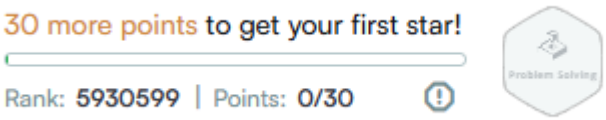
I am here to

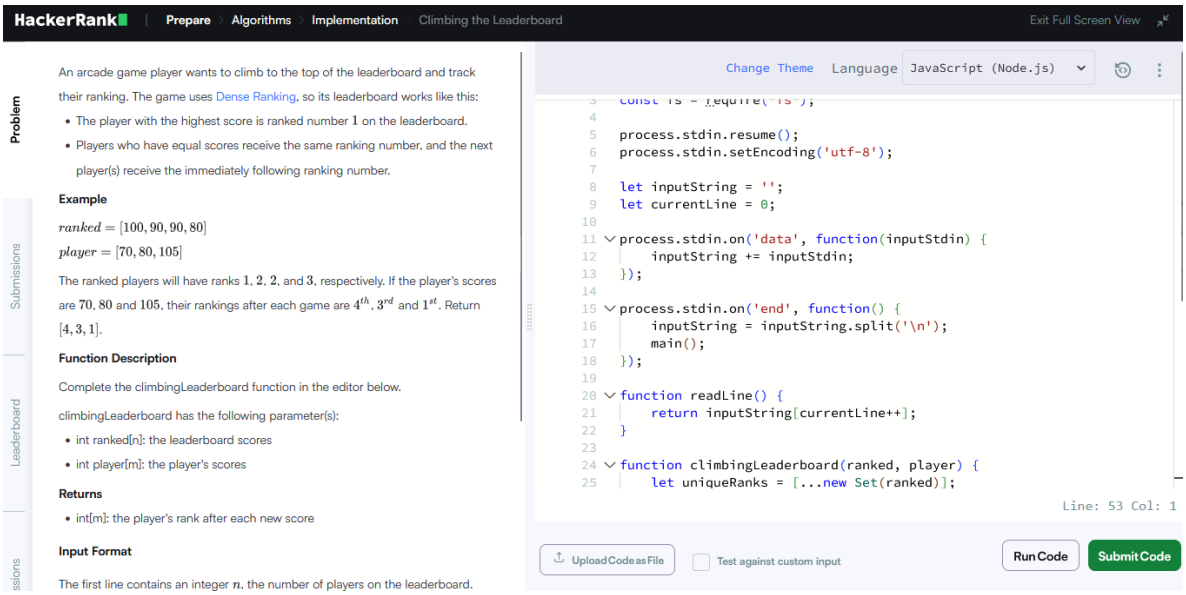
Prepare for Job Interviews

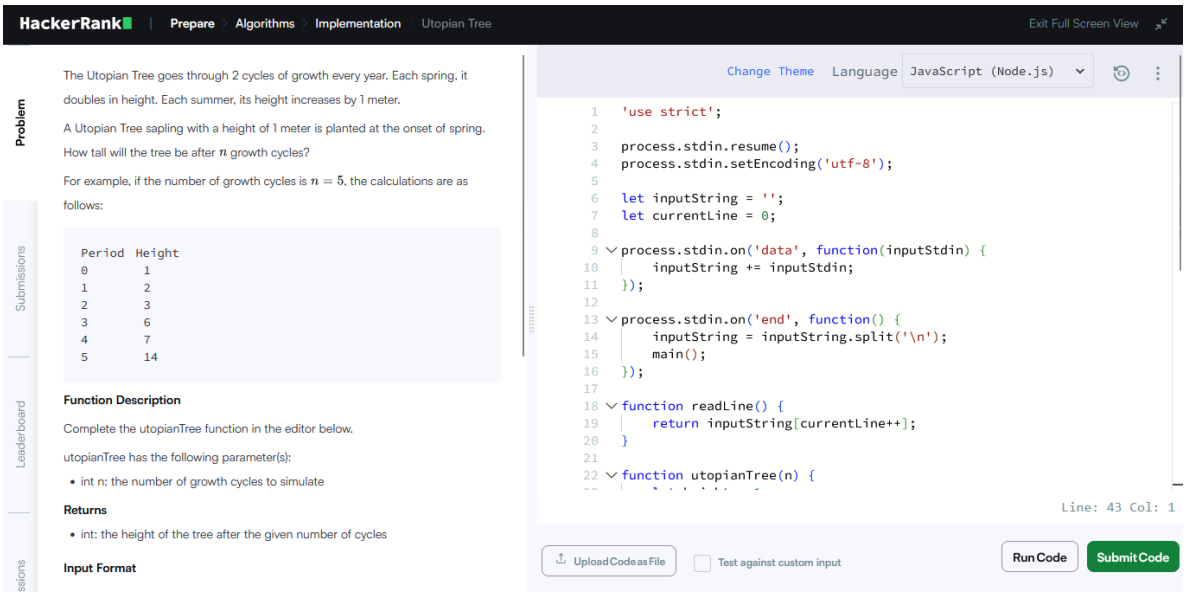
Learn & Compete with Others




HackerRank
Prepare
Certify
Compete
Apply
Search
Prepare > Problem Solving
Problem Solving
30 more points to get your first star!
Rank: 5930599 | Points: 0/30
Algorithms | Data Structures
Solve Me First
Easy, Problem Solving (Basic), Max Score: 1, Success Rate: 97.64%
This is an easy challenge to help you start coding in your favorite languages!
Solve Challenge
Simple Array Sum
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 94.67%
Solve Challenge
Compare the Triplets
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 95.96%
Solve Challenge
A Very Big Sum
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 98.84%
Solve Challenge
STATUS
Solved
Unsolved
SKILLS
Problem Solving (Intermediate)
Problem Solving (Advanced)
Problem Solving (Basic)
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Warmup

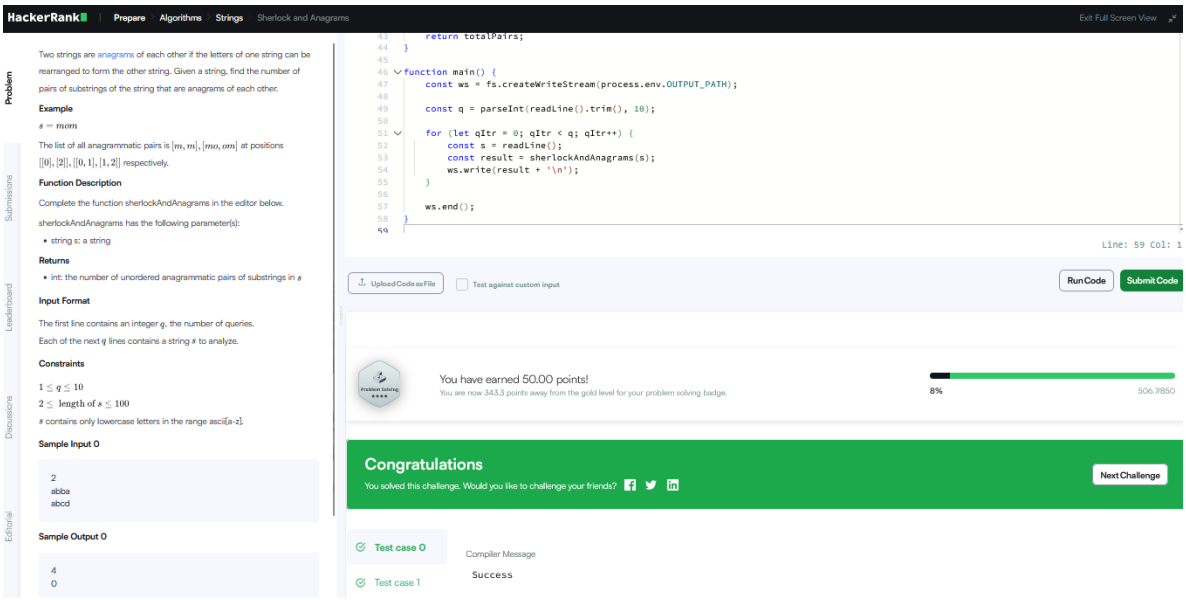

30 more points to get your first star!
Rank: 5930599 | Points: 0/30
Problem Solving

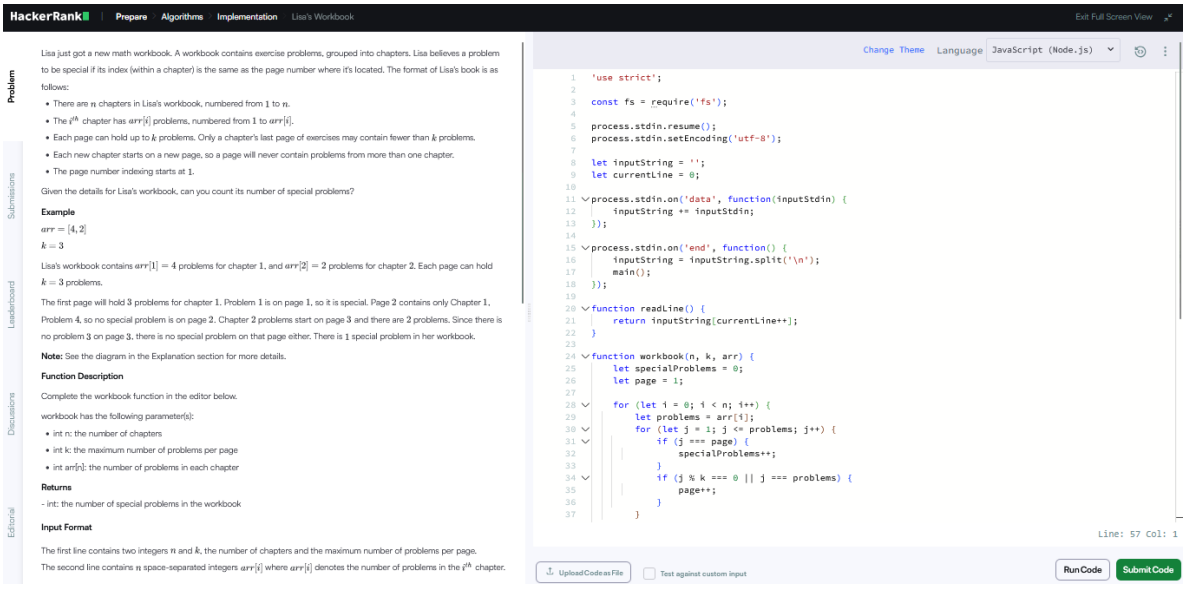

HackerRank
Prepare
Algorithms
Implementation
Climbing the Leaderboard
Exit Full Screen View
Problem
Submissions
Leaderboard
An arcade game player wants to climb to the top of the leaderboard and track their ranking. The game uses Dense Ranking, so its leaderboard works like this:
The player with the 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 4th, 3rd and 1st. 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.
Change Theme
Language
JavaScript (Node.js)
process.stdin.resume();
process.stdin.setEncoding('utf-8');
let inputString = '';
let currentLine = 0;
process.stdin.on('data', function(inputStdin) {
inputString += inputStdin;
});
process.stdin.on('end', function() {
inputString = inputString.split('\n');
main();
});
function readLine() {
return inputString[currentLine++];
}
function climbingLeaderboard(ranked, player) {
let uniqueRanks = [...new Set(ranked)];
Line: 53 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code




```
'use strict';

const fs = require('fs');

process.stdin.resume();
process.stdin.setEncoding('utf-8');

let inputString = '';
let currentLine = 0;

process.stdin.on('data', function (inputStdin) {
    inputString += inputStdin;
});

process.stdin.on('end', function () {
    inputString = inputString.split('\n');
    main();
});

function readLine() {
    return inputString[currentLine++];
}

function sherlockAndAnagrams(s) {
    let substrCount = new Map();
    let totalPairs = 0;

    for (let start = 0; start < s.length; start++) {
        for (let end = start; end < s.length; end++) {
            let subStr = s.substring(start, end + 1);
            let sortedSubStr = subStr.split('').sort().join('');

            substrCount.set(sortedSubStr, (substrCount.get(sortedSubStr) || 0) + 1);
        }
    }

    for (let count of substrCount.values()) {
```

```
    for (let count of substrCount.values()) {
        if (count > 1) {
            totalPairs += (count * (count - 1)) / 2;
        }
    }

    return totalPairs;
}

function main() {
    const ws = fs.createWriteStream(process.env.OUTPUT_PATH);

    const q = parseInt(readLine().trim(), 10);

    for (let qItr = 0; qItr < q; qItr++) {
        const s = readLine();
        const result = sherlockAndAnagrams(s);
        ws.write(result + '\n');
    }
```





- $1 \leq n, k, arr[i] \leq 100$

Sample Input

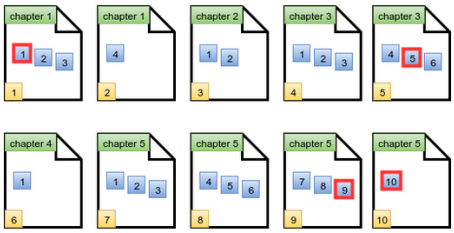
```
STDIN       Function
-----       --------
5 3         n = 5, k = 3
4 2 6 1 10  arr = [4, 2, 6, 1, 10]
```

Sample Output

```
4
```

Explanation

The diagram below depicts Lisa's workbook with $n = 5$ chapters and a maximum of $k = 3$ problems per page. Special problems are outlined in red, and page numbers are in yellow squares.



There are 4 special problems and thus we print the number 4 on a new line.

```
}

function workbook(n, k, arr) {
    let specialProblems = 0;
    let page = 1;

    for (let i = 0; i < n; i++) {
        let problems = arr[i];
        for (let j = 1; j <= problems; j++) {
            if (j === page) {
                specialProblems++;
            }
            if (j % k === 0 || j === problems) {
                page++;
            }
        }
    }

    return specialProblems;
}

function main() {
    const ws = fs.createWriteStream(process.env.OUTPUT_PATH);

    const firstMultipleInput = readLine().replace(/\s+$/g, '').split(' ');

    const n = parseInt(firstMultipleInput[0], 10);
    const k = parseInt(firstMultipleInput[1], 10);
```

Line: 57 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 25.00 points!
You are now 93.3 points away from the gold level for your problem solving badge.
75% 756.7/850

HackerRank | Prepare | Algorithms | Implementation | Manasa and Stones

Returns

- int[]: all possible values of the last stone, sorted ascending

Input Format

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

Each test case contains 3 lines:
- The first line contains n, the number of non-zero stones found.
- The second line contains a, one possible difference
- The third line contains b, the other possible difference.

Constraints

- $1 \leq T \leq 10$
- $1 \leq n, a, b < 10^3$

Sample Input

```
STDIN  Function
-----  --------
2      T = 2 (test cases)
3      n = 3 (test case 1)
1      a = 1
2      b = 2
4      n = 4 (test case 2)
10     a = 10
100    b = 100
```

Sample Output

```
2 3 4
30 120 210 300
```

Explanation

With differences 1 and 2, all possible series for the first test case are given below:

1. 0,1,**2**
2. 0,1,**3**
3. 0,2,**3**
4. 0,2,**4**

```
}

function main() {
    const ws = fs.createWriteStream(process.env.OUTPUT_PATH);

    const T = parseInt(readLine().trim(), 10);

    for (let TItr = 0; TItr < T; TItr++) {
        const n = parseInt(readLine().trim(), 10);
        const a = parseInt(readLine().trim(), 10);
        const b = parseInt(readLine().trim(), 10);

        const result = stones(n, a, b);

        .join(' ') + '\n');
```

Problem Solving

Congrats!

You have earned your 5th star.

Continue

Line: 62 Col: 1

Run Code | Submit Code

You have earned 30.00 points!
You are now 1333.3 points away from the 6th star for your problem solving badge.
1% 866.7/2200

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0 | Compiler Message

Problem

Submissions

Leaderboard

Discussions

Editorial

Your local library needs your help! Given the expected and actual return dates for a library book, create a program that calculates the fine (if any). The fee structure is as follows:

1. If the book is returned on or before the expected return date, no fine will be charged (i.e.: $fine = 0$).
2. If the book is returned after the expected return day but still within the same calendar month and year as the expected return date, $fine = 15$ Hackos $\times$ (the number of days late).
3. If the book is returned after the expected return month but still within the same calendar year as the expected return date, the $fine = 500$ Hackos $\times$ (the number of months late).
4. If the book is returned after the calendar year in which it was expected, there is a fixed fine of 10000 Hackos.

Charges are based only on the least precise measure of lateness. For example, whether a book is due January 1, 2017 or December 31, 2017, if it is returned January 1, 2018, that is a year late and the fine would be $10,000$ Hackos.

Example

$d1, m1, y1 = 14, 7, 2018$
$d2, m2, y2 = 5, 7, 2018$

The first values are the return date and the second are the due date. The years are the same and the months are the same. The book is $14 - 5 = 9$ days late. Return $9 * 15 = 135$.

Function Description

Complete the libraryFine function in the editor below.

libraryFine has the following parameter(s):

- d1, m1, y1: returned date day, month and year, each an integer
- d2, m2, y2: due date day, month and year, each an integer

Returns

- int: the amount of the fine or 0 if there is none

Input Format

The first line contains 3 space-separated integers, $d1, m1, y1$, denoting the respective day, $month$, and $year$ on which the book was returned.
The second line contains 3 space-separated integers, $d2, m2, y2$, denoting the respective day, $month$, and $year$ on which the book was due to be returned.

Constraints

- $1 \leq d1, d2 \leq 31$
- $1 \leq m1, m2 \leq 12$

Change Theme Language JavaScript (Node.js)

```
'use strict';

const fs = require('fs');

process.stdin.resume();
process.stdin.setEncoding('utf-8');

let inputString = '';
let currentLine = 0;

process.stdin.on('data', function(inputStdin) {
    inputString += inputStdin;
});

process.stdin.on('end', function() {
    inputString = inputString.split('\n');

    main();
});

function readLine() {
    return inputString[currentLine++];
}

/*
 * Complete the 'libraryFine' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER d1
 *  2. INTEGER m1
 *  3. INTEGER y1
 *  4. INTEGER d2
 *  5. INTEGER m2
 *  6. INTEGER y2
 */

```

Line: 48 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

HackerRank | Prepare / Algorithms / Implementation / Library Fine Exit Full Screen View

Problem

Submissions

Leaderboard

Discussions

Editorial

Your local library needs your help! Given the expected and actual return dates for a library book, create a program that calculates the fine (if any). The fee structure is as follows:

1. If the book is returned on or before the expected return date, no fine will be charged (i.e.: $fine = 0$).
2. If the book is returned after the expected return day but still within the same calendar month and year as the expected return date, $fine = 15$ Hackos $\times$ (the number of days late).
3. If the book is returned after the expected return month but still within the same calendar year as the expected return date, the $fine = 500$ Hackos $\times$ (the number of months late).
4. If the book is returned after the calendar year in which it was expected, there is a fixed fine of 10000 Hackos.

Charges are based only on the least precise measure of lateness. For example, whether a book is due January 1, 2017 or December 31, 2017, if it is returned January 1, 2018, that is a year late and the fine would be $10,000$ Hackos.

Example

$d1, m1, y1 = 14, 7, 2018$
$d2, m2, y2 = 5, 7, 2018$

The first values are the return date and the second are the due date. The years are the same and the months are the same. The book is $14 - 5 = 9$ days late. Return $9 * 15 = 135$.

Function Description

Complete the libraryFine function in the editor below.

libraryFine has the following parameter(s):

- d1, m1, y1: returned date day, month and year, each an integer
- d2, m2, y2: due date day, month and year, each an integer

Returns

- int: the amount of the fine or 0 if there is none

Input Format

The first line contains 3 space-separated integers, $d1, m1, y1$, denoting the respective day, $month$, and $year$ on which the book was returned.
The second line contains 3 space-separated integers, $d2, m2, y2$, denoting the respective day, $month$, and $year$ on which the book was due to be returned.

Constraints

- $1 \leq d1, d2 \leq 31$
- $1 \leq m1, m2 \leq 12$

```
function libraryFine(d1, m1, y1, d2, m2, y2) {
    if (y1 < y2 || (y1 === y2 && m1 < m2) || (y1 === y2 && m1 === m2 && d1 <= d2)) {
        return 0;
    } else if (y1 > y2) {
        return 10000;
    } else if (m1 > m2) {
        return 500 * (m1 - m2);
    } else {
        return 15 * (d1 - d2);
    }
}

function main() {
    const ws = fs.createWriteStream(process.env.OUTPUT_PATH);

    const firstMultipleInput = readLine().replace(/\s+$/g, '').split(' ');

    const d1 = parseInt(firstMultipleInput[0], 10);

```

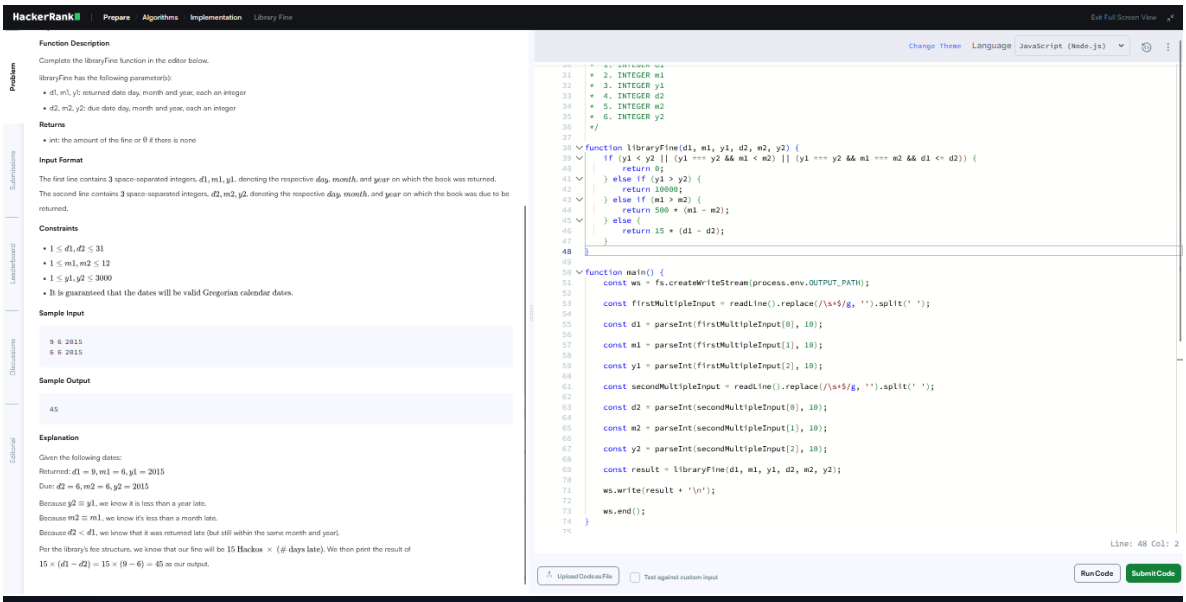
Line: 48 Col: 2

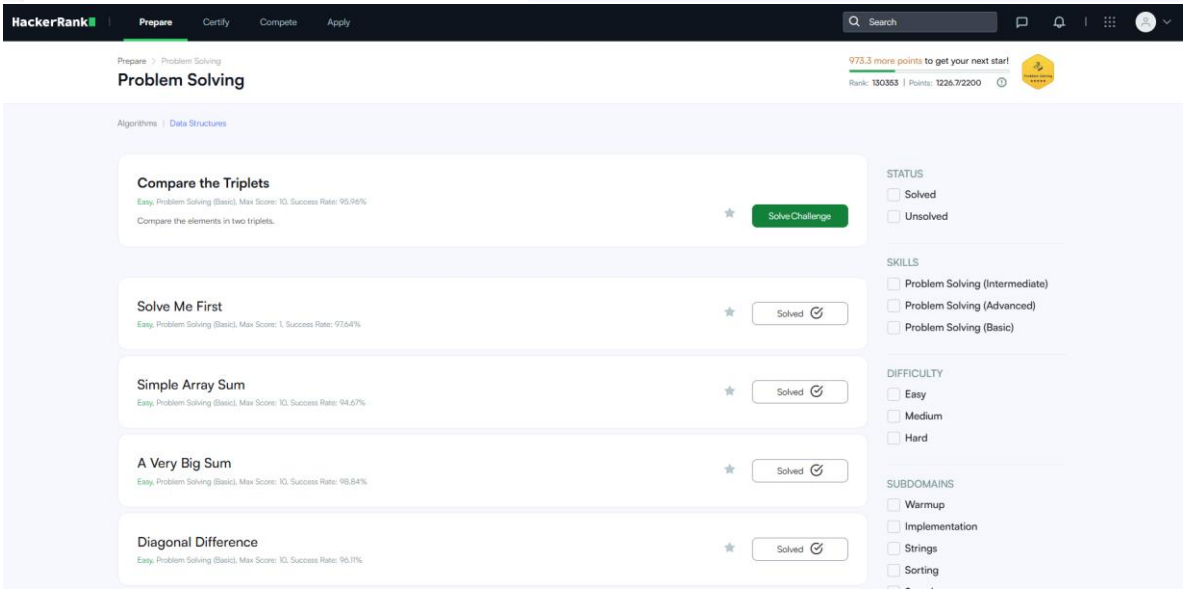
Upload Code as File Test against custom input Run Code Submit Code

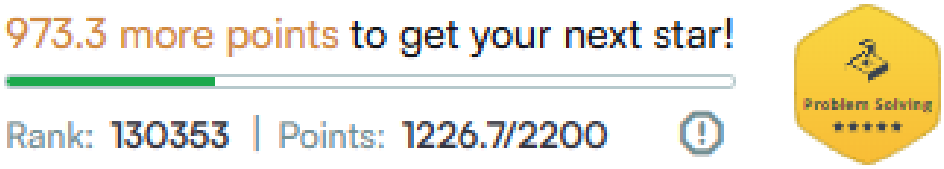
You have earned 15.00 points!
You are now 1188.3 points away from the 6th star for your problem solving badge.
12% 1011.7/2200

Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge

Test case 0 Compiler Message


HackerRank
Prepare
Algorithms
Implementation
Library Fine
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
Editorial
Function Description
Complete the libraryFine function in the editor below.
libraryFine has the following parameter(s):
• d1, m1, y1: returned date day, month and year, each an integer
• d2, m2, y2: due date day, month and year, each an integer
Returns
• int: the amount of the fine or 0 if there is none
Input Format
The first line contains 3 space-separated integers, d1, m1, y1, denoting the respective day, month, and year on which the book was returned.
The second line contains 3 space-separated integers, d2, m2, y2, denoting the respective day, month, and year on which the book was due to be returned.
Constraints
• 1 ≤ d1, d2 ≤ 31
• 1 ≤ m1, m2 ≤ 12
• 1 ≤ y1, y2 ≤ 3000
• It is guaranteed that the dates will be valid Gregorian calendar dates.
Sample Input
9 6 2015
6 6 2015
Sample Output
45
Explanation
Given the following dates:
Returned: d1 = 9, m1 = 6, y1 = 2015
Due: d2 = 6, m2 = 6, y2 = 2015
Because y2 ≡ y1, we know it is less than a year late.
Because m2 ≡ m1, we know it's less than a month late.
Because d2 < d1, we know that it was returned late (but still within the same month and year).
Per the library's fee structure, we know that our fine will be 15 Hackos × (# days late). We then print the result of
15 × (d1 − d2) = 15 × (9 − 6) = 45 as our output.
Change Theme
Language
JavaScript (Node.js)
Line: 48 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code


HackerRank
Prepare
Certify
Compete
Apply
Search
Prepare > Problem Solving
Problem Solving
973.3 more points to get your next star!
Rank: 130353 | Points: 1226.7/2200
Algorithms | Data Structures
Compare the Triplets
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 95.96%
Compare the elements in two triplets.
Solve Challenge
Solve Me First
Easy, Problem Solving (Basic), Max Score: 1, Success Rate: 97.64%
Solved
Simple Array Sum
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 94.67%
Solved
A Very Big Sum
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 98.84%
Solved
Diagonal Difference
Easy, Problem Solving (Basic), Max Score: 10, Success Rate: 96.11%
Solved
STATUS
Solved
Unsolved
SKILLS
Problem Solving (Intermediate)
Problem Solving (Advanced)
Problem Solving (Basic)
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Warmup
Implementation
Strings
Sorting


973.3 more points to get your next star!
Rank: 130353 | Points: 1226.7/2200
Problem Solving