ето ги и задачките от вчера

Dimo изпрати Днес в 8:49 ч.

Days of Week -> [https://pastebin.com/zjEwTkYa](https://l.facebook.com/l.php?u=https%3A%2F%2Fpastebin.com%2FzjEwTkYa%3Ffbclid%3DIwAR0P35ZyhKATnKFYZ5PRZmapixbcdhaTjEUEBvhSGRlLH33VIGkC08KVFCU&h=AT1E5nMT6ZfHsE0I3pwlv43uDJSqNwiDi_YQXC5ou8mr_8LheXjc2-DDzle0An8w1cA1QEEJn9YFUiSq7PN6sirPkmw-98PRAhFs5aQSuyoS1TSttfOBoTH-_5oeO0za8rqZ1w)

Print Numbers in Reverse -> [https://pastebin.com/uPD2rKaa](https://l.facebook.com/l.php?u=https%3A%2F%2Fpastebin.com%2FuPD2rKaa%3Ffbclid%3DIwAR1NvMtbZ-nymS74QTh1WSvwlaI00L6lt3OjUeqtlsjmrgYsWM3vAnwDX_o&h=AT1E5nMT6ZfHsE0I3pwlv43uDJSqNwiDi_YQXC5ou8mr_8LheXjc2-DDzle0An8w1cA1QEEJn9YFUiSq7PN6sirPkmw-98PRAhFs5aQSuyoS1TSttfOBoTH-_5oeO0za8rqZ1w)

Reverse Array of Strings -> [https://pastebin.com/aEMJpZwx](https://l.facebook.com/l.php?u=https%3A%2F%2Fpastebin.com%2FaEMJpZwx%3Ffbclid%3DIwAR1cjYuPdDCJtqKrUcSGMUpt0ewfq6-2Vw42bgPB3mCl7DBokk1g4OOpNHU&h=AT1E5nMT6ZfHsE0I3pwlv43uDJSqNwiDi_YQXC5ou8mr_8LheXjc2-DDzle0An8w1cA1QEEJn9YFUiSq7PN6sirPkmw-98PRAhFs5aQSuyoS1TSttfOBoTH-_5oeO0za8rqZ1w)

Even Odd Subtraction -> [https://pastebin.com/MnijWLj9](https://l.facebook.com/l.php?u=https%3A%2F%2Fpastebin.com%2FMnijWLj9%3Ffbclid%3DIwAR3m0lYhnpSlZ3u_UmB98vEZTxTjWtcqwRoO6O2zxPAU-flc8rXJnRqNpig&h=AT1E5nMT6ZfHsE0I3pwlv43uDJSqNwiDi_YQXC5ou8mr_8LheXjc2-DDzle0An8w1cA1QEEJn9YFUiSq7PN6sirPkmw-98PRAhFs5aQSuyoS1TSttfOBoTH-_5oeO0za8rqZ1w)

Equal Arrays -> [https://pastebin.com/TqtJfRdw](https://pastebin.com/TqtJfRdw?fbclid=IwAR183F_qq8TpOVkN5rMd6SKw_WIBU5-CaV3sRZT4j1dRU5hwcowv84JNQlA)

String exerciseLink = daysOfWeek -> { System.OutPrintln(daysOfWeek) }\*\*\*

Velina

**Velina изпрати Днес в 10:58 ч.**

Колеги имам въпрос сега изгледах цялата лекция от вчера и и решавам 6та задача за проверка дали два масива са еднакви. Мисля, че не е обсъждано, но решението не покрива случай в който втория масив е по-дълъг, но първите му елементи са еднакви с тези от първия. В този случай отново дава че са еднакви. Но и джъдж това не го интересува и дава 100 точки въпреки че този вариянт не е покрит.

В следващи задачки и този случай ще се има предвид

https://pastebin.com/iqsTXwzU?fbclid=IwAR0OVZxF-YzbcgLUzOUKjkfcLCBdsjDJBOkHj5V7i5jmYOY1oCFooE7eVyM

import java.util.Scanner;

public class FLab07\_07CondenseArrayToNumber {

    static int[] toIntArray(String input) {

        String[] splitInput = input.split(" ");

        int[] array = new int[splitInput.length];

        for (int i = 0; i < array.length; i++) {

            array[i] = Integer.parseInt(splitInput[i]);

        }

        return array;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int[] array = toIntArray(scan.nextLine());

        if (array.length == 1) {

            System.out.printf("%d", array[0]);

        } else {

            // \*\*\*\*\*\*\*\*\*\*\*\*\*\* ver.01: No recursion, only 2 arrays are used. Allocated only once in the RAM;

            /\*

            int[] condensed = new int[array.length];

            for (int i = 0; i < array.length; i++) {

                for (int j = 0; j < array.length - 1; j++) {

                    condensed[j] = array[j] + array[j + 1];

                }

                //Updating array[] and adding ZEROs in the tail:

                for (int j = 0; j < array.length; j++) {

                    if (j >= array.length - 1 - i) {

                        array[j] = 0;

                    } else {

                        array[j] = condensed[j];

                    }

                }

            }

            System.out.println(condensed[0]);

            \*/

            // \*\*\*\*\*\*\*\*\*\* ver02: Solution via recursion:

            while (array.length > 1){

                int[] condensed = new int[array.length-1];

                for (int i = 0; i < array.length-1; i++) {

                    condensed[i] = array[i] + array[i+1];

                }

                //Making the two so far independent array to point to very same place in the heap;

                array = condensed;

                // the next roll of the loop will create a brand new stack with another condensed[]

                // array that will be one element shorter from the current one...

                //One we are done, we will sill keep the reference of the most resent state of condensed[]

                // via the array[] variable;

            }

            System.out.println(array[0]);

        }

    }

}

https://pastebin.com/jaSjUmBp?fbclid=IwAR2NW-DrxlyPGAuqhzQRdLy\_IpUvAA3gI7ENTpFz93SVLZFbgU3g4SFrP3c

import java.util.Scanner;

public class CondenseArrayToNumber {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        String input = scanner.nextLine();

        String[] numbersAsStrings = input.split(" ");

        int[] numbers = new int[numbersAsStrings.length];

        for (int i = 0; i < numbers.length; i++) {

            numbers[i] = Integer.parseInt(numbersAsStrings[i]);

        }

        while (numbers.length > 1) {

            int[] numbersCondensed = new int[numbers.length - 1];

            for (int i = 0; i < numbersCondensed.length; i++) {

                numbersCondensed[i] = numbers[i] + numbers[i + 1];

            }

            numbers = numbersCondensed;

        }

        System.out.println(numbers[0]);

    }

}

https://pastebin.com/m567UVLv?fbclid=IwAR28UauiQvGVAJlSmFcPsK39pkiCxIjP-ZVr1ff2usNwgrfs-txYIacT\_ds

7.  Condense Array to Number

Write a program to read an array of integers and condense them by summing adjacent couples of elements until a single integer is obtained. For example, if we have 3 elements {2, 10, 3}, we sum the first two and the second two elements and obtain {2+10, 10+3} = {12, 13}, then we sum again all adjacent elements and obtain {12+13} = {25}.

 Input         Output      Comments

 2 10 3         25         2 10 3 -> 2+10 10+3 -> 12 13 -> 12 + 13 -> 25

 5 0 4 1 2      35         5 0 4 1 2 -> 5+0 0+4 4+1 1+2 -> 5 4 5 3 -> 5+4 4+5 5+3 -> 9 9 8 -> 9+9 9+8 -> 18 17 -> 18+17 -> 35

 1              1          1 is already condensed to number

Hints

While we have more than one element in the array nums[], repeat the following:

• Allocate a new array condensed[] of size nums.Length-1.

• Sum the numbers from nums[] to condensed[]:

     o  condensed[i] = nums[i] + nums[i+1]

• nums[] = condensed[]

package Arrays.Exercise;

import java.util.Arrays;

import java.util.Scanner;

public class Test5 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int[] number = Arrays.stream(scanner.nextLine().split(" ")).

                mapToInt(value -> Integer.parseInt(value)).toArray();

        int countCalculations = number.length - 1;

        boolean isCompleteCalculations = false;

        int result = 0;

        while (countCalculations > 0) {

            int[] condensed1 = new int[number.length - 1];

            for (int i = 0; i < number.length - 1; i++) {

                condensed1[i] = number[i] + number[i + 1];

                number[i] = condensed1[i];

                if (i == condensed1.length - 1) {

                    countCalculations--;

                    if (countCalculations == 0) {

                        isCompleteCalculations = true;

                        result = condensed1[i];

                        break;

                    }

                }

            }

            if (isCompleteCalculations) {

                break;

            }

            int[] condensed2 = new int[condensed1.length - 1];

            for (int j = 0; j < condensed1.length - 1; j++) {

                condensed2[j] = condensed1[j] + condensed1[j + 1];

                condensed1[j] = condensed2[j];

                if (j == condensed2.length - 1) {

                    countCalculations--;

                    if (countCalculations == 0) {

                        isCompleteCalculations = true;

                        result = condensed2[j];

                        break;

                    }

                }

            }

            if (isCompleteCalculations) {

                break;

            }

            int[] condensed3 = new int[condensed2.length - 1];

            for (int k = 0; k < condensed2.length - 1; k++) {

                condensed3[k] = condensed2[k] + condensed2[k + 1];

                condensed2[k] = condensed3[k];

                if (k == condensed3.length - 1) {

                    countCalculations--;

                    if (countCalculations == 0) {

                        isCompleteCalculations = true;

                        result = condensed3[k];

                        break;

                    }

                }

            }

            if (isCompleteCalculations) {

                break;

            }

            int[] condensed4 = new int[condensed3.length - 1];

            for (int l = 0; l < condensed3.length - 1; l++) {

                condensed4[l] = condensed3[l] + condensed3[l + 1];

                condensed3[l] = condensed4[l];

                if (l == condensed4.length - 1) {

                    countCalculations--;

                    if (countCalculations == 0) {

                        isCompleteCalculations = true;

                        result = condensed4[l];

                        break;

                    }

                }

            }

            if (isCompleteCalculations) {

                break;

            }

        }

        if (isCompleteCalculations) {

            System.out.println(result);

        } else {

            result = number[number.length - 1];

            System.out.println(result);

        }

    }

}