**Задания**

**Занятие №6**

1. **Напечатать ряд из повторяющихся чисел 20 в виде: 20 20 20 20 20 20 20 20 20 20 20 20 20**

Код:

let str: string = '';

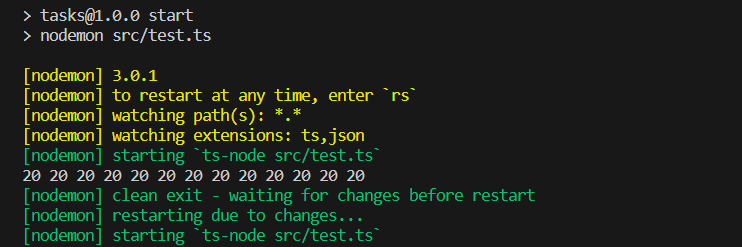
for (let i: number = 0; i < 13; i++) {

    str += '20 '

}

console.log(str)

Результат:



1. **Вычислить факториал числа n (n!). Факториал числа n равен произведению чисел от1 до n. Причем, 0!=1.**

Код:

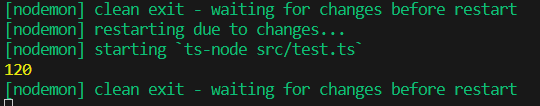
function factorial(n: number): number {

    return (n != 1) ? n \* factorial(n - 1) : 1;

}

console.log(factorial(5))

Результат:



1. **Напечатать «столбиком» все целые числа от 20 до 35.**

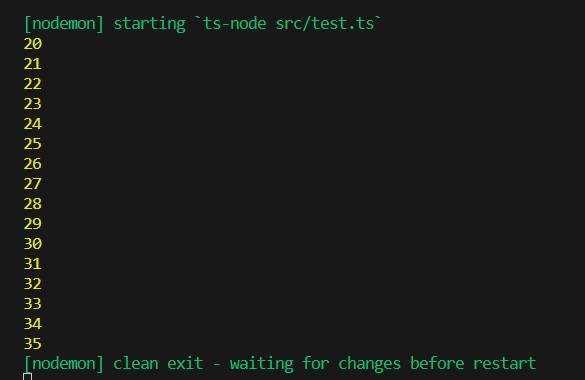
Код:

for (let i: number = 20; i <= 35; i++) {

    console.log(i)

}

Результат:



1. **Напечатать «столбиком» квадраты всех целых чисел от 10 до b (значение b вводится с клавиатуры; b≥10).**

Код:

function printSquareNum(b: number): void {

    if (b >= 10) {

        for (let i: number = 10; i <= b; i++) {

            console.log(i \* i)

        }

    } else {

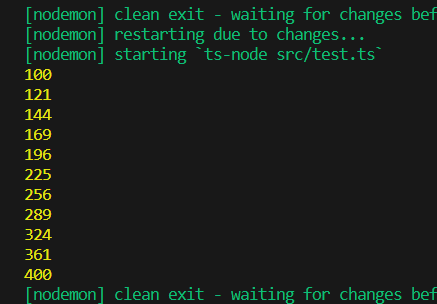
        console.log(`${b} does not match condition ${b} >= 10`)

    }

}

printSquareNum(prompt('Введите число: ', 0))

Результат:



1. **Напечатать «столбиком» третьи степени всех целых чисел от a до 50 (значение a вводится с клавиатуры; a ≤ 50).**

Код:

function printPowNum(a: number): void {

    if (a <= 50) {

        for (let i: number = a; i <= 50; i++) {

            console.log(Math.pow(i, 3))

        }

    } else {

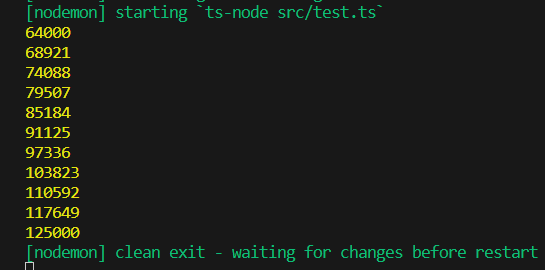
        console.log(`${a} does not match condition ${a} <= 50`)

    }

}

printPowNum(prompt('Введите число: ', 0))

Результат:



1. **Одна штука некоторого товара стоит 20,4 руб. Напечатать таблицу стоимости 2, 3, ..., 20 штук этого товара.**

Код:

function printTablePrice(): void {

    const arrProduct: number[] = []

    const price: number = 20.4

    function Product(count: number): void {

        this.count = count;

        this.price = Number((count \* price).toFixed(1));

    }

    for (let i = 1; i <= 20; i++) {

        arrProduct.push(new Product(i))

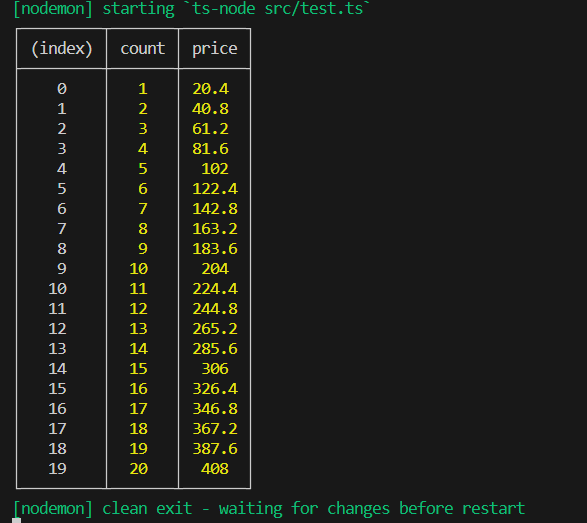
    }

    console.table(arrProduct)

}

printTablePrice()

Результат:



1. **Напечатать таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, ..., 10 фунтов (1 фунт = 453 г).**

Код:

function printTableWeight(): void {

    const arrComparison: number[] = []

    const poundValue: number = 0.453

    function Comparison(pound: number): void {

        this.pounds = pound;

        this.kilograms = Number((pound \* poundValue).toFixed(3));

    }

    for (let i = 1; i <= 10; i++) {

        arrComparison.push(new Comparison(i))

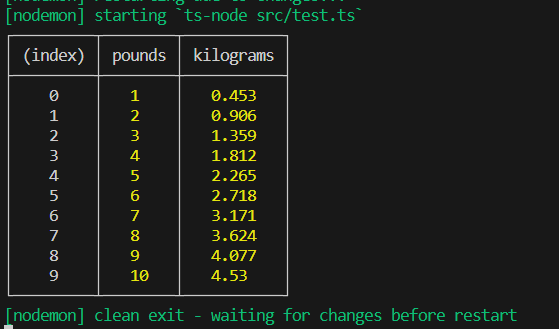
    }

    console.table(arrComparison)

}

printTableWeight()

Результат:



1. **Напечатать таблицу соответствия расстояний в дюймах расстояниям в сантиметрах для значений 10, 11, ..., 22 дюйма (1 дюйм = 25,4 мм).**

Код:

function printTableDistance(): void {

    const arrComparison: number[] = []

    const inchValue: number = 2.54

    function Comparison(inch: number): void {

        this.inchs = inch;

        this.centimetres = inch \* inchValue;

    }

    for (let i = 10; i <= 22; i++) {

        arrComparison.push(new Comparison(i))

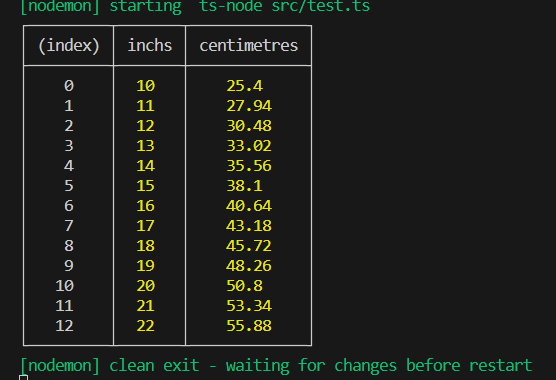
    }

    console.table(arrComparison)

}

printTableDistance()

Результат:



1. **Считая, что Земля – идеальная сфера с радиусом r 6350 км, определить расстояние до линии горизонта от точки с высотой над Землей, равной 1, 2, ..., 10 км.**

Код:

function printHorizonLine(): void {

    const radius: number = 6350;

    for (let h: number = 1; h <= 10; h++) {

        const horizonLine: number = Math.floor(Math.sqrt(2 \* h \* radius))

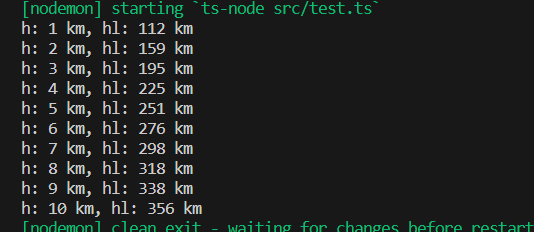
        console.log(`h: ${h} km, hl: ${horizonLine} km`)

    }

}

printHorizonLine()

Результат:



1. **Напечатать таблицу перевода 1, 2, ..., 20 долларов США в рубли по текущему курсу (значение курса вводится с клавиатуры).**

Код:

function prinTableExchangeRate(rate: number): void {

    const arrRate: number[] = []

    function Rates(value: number): void {

        this.dollars = value;

        this.rubles = Number((value \* rate).toFixed(2))

    }

    for (let i: number = 1; i <= 20; i++) {

        arrRate.push(new Rates(i))

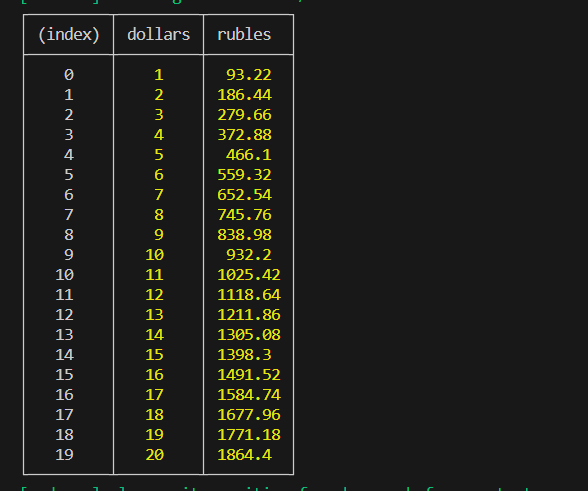
    }

    console.table(arrRate)

}

prinTableExchangeRate(prompt('Enter the current exchange rate: ', 0))

Результат:



1. **Распечатать в «столбик» таблицу умножения на 7.**

Код:

function prinMultiplicationTable(value: number): void {

    type T = string[]

    const arrValues: T[] = []

    for (let i: number = 1; i <= 10; i++) {

        arrValues.push([`${value} \* ${i} = ${value \* i}`])

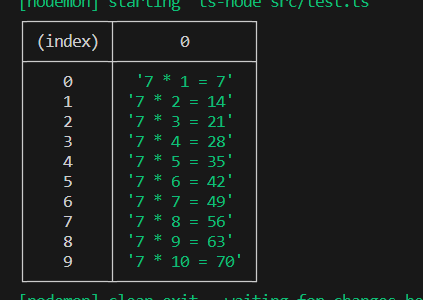
    }

    console.table(arrValues)

}

prinMultiplicationTable(7)

Результат:



1. **Распечатать в «столбик» таблицу умножения на 9.**

Код:

function prinMultiplicationTable(value: number): void {

    type T = string[]

    const arrValues: T[] = []

    for (let i: number = 1; i <= 10; i++) {

        arrValues.push([`${value} \* ${i} = ${value \* i}`])

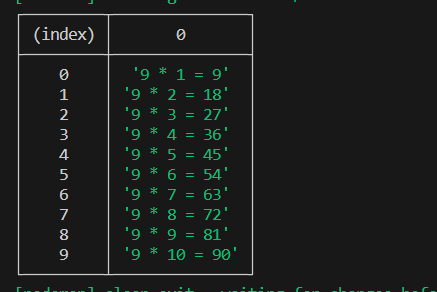
    }

    console.table(arrValues)

}

prinMultiplicationTable(9)

Результат:

****

1. **Спортсмен, начав тренировки, пробежал в первый день n километров. В каждый следующий день он увеличивает пробег на 10%. Сколько километров пробежит спортсмен на 7 день тренировок.**

Код:

function getCountKillometrs(count: number): void {

    let distance = count;

    for (let i: number = 2; i <= 7; i++) {

        distance = distance \* 110 / 100

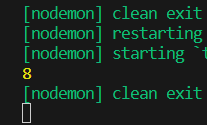
    }

    console.log(Math.floor(distance))

}

getCountKillometrs(5)

Результат:



1. **Вычислить S = sin x + sin 2x + sin 3x + ….+ sin nx**

Код:

function getSumSin(n: number, x: number): void {

    let sum: number = 0;

    for (let i: number = 1; i <= n; i++) {

        sum = sum + Math.sin(n \* x)

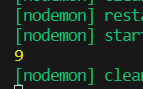
    }

    console.log(Math.floor(sum))

}

getSumSin(10, 2)

Результат:



1. **Вычислить S = 1 + 11 + 111 + 1111 + ….+ 11111…111 n**

Код:

function getSummation(n: number): number {

    let sum: number = 0;

    let value: number = 1;

    for (let i: number = 1; i <= n; i++) {

        sum = sum + value;

        value = (value \* 10) + 1

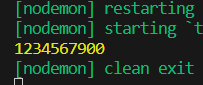
    }

    return sum

}

console.log(getSummation(10))

Результат:



1. **Составить программу вычисления значений функции F(x) на отрезке [a,b] c шагом h. Результат представить в виде таблицы, первый столбец которой – значение аргумента, второй - соответствующее значение функции:**

Код:

type T = number[]

function getFunctionValue(a: number, b: number, h: number): T[] {

    const arrValues: T[] = []

    function Comparison(x: number, func: number): void {

        this.argument = x

        this.function = func

    }

    function firstFunc(x: number): number {

        return Number((x - Math.sin(x)).toFixed(3))

    }

    function secondFunc(x: number): number {

        return Number((Math.sin(Math.pow(x, 2))).toFixed(3))

    }

    function thirdFunc(x: number): number {

        return Number((Math.sqrt(x) \* Math.cos(Math.pow(x, 2))).toFixed(3))

    }

    for (let x: number = a; x <= b; x += h) {

        arrValues.push(new Comparison(x, firstFunc(x)))

        arrValues.push(new Comparison(x, secondFunc(x)))

        arrValues.push(new Comparison(x, thirdFunc(x)))

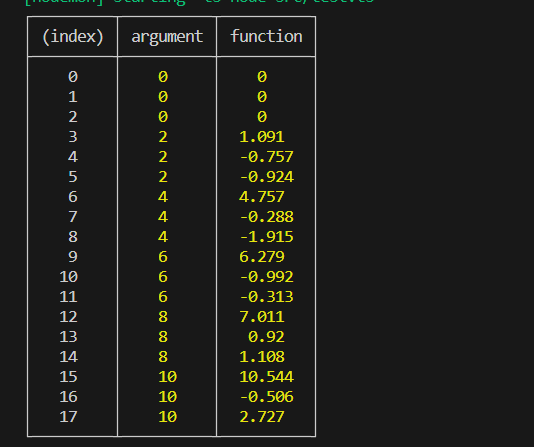
    }

    return arrValues

}

console.table(getFunctionValue(0, 10, 2))

Результат:

****

1. **Найти наименьший номер элемента последовательности, для которого выполняется условие М. Вывести на экран этот номер.**

Код:

function findSmallN(epsilon: number): number {

    let n = 1;

    let a\_n = getA\_n(n);

    function getA\_n(n: number): number {

        return 1 / Math.pow(n + 1, 2)

    }

    while (a\_n >= epsilon) {

        n++;

        a\_n = getA\_n(n);

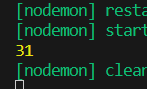
    }

    return n

}

console.log(findSmallN(0.001))

Результат:



1. **Сберегательный банк начисляет 7% годовых, т.е. через год вклад увеличивается без участия вкладчика. Какой станет сумма К рублей, положенная в банк на N лет?**

Код:

function calculateTotalAmount(value: number, years: number): number {

    let sum = value;

    for (let y: number = 1; y <= years; y++) {

        sum = sum + (sum \* 0.07)

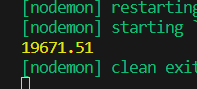
    }

    return Number(sum.toFixed(2))

}

console.log(calculateTotalAmount(10000, 10))

Результат:

****

1. **Вычислить сумму:**

Код:

function calculateSum(n: number): number {

    let sum = 1;

    let numerator = 0;

    let denominator = 1;

    let value = 2;

    for (let i = 1; i <= n; i ++) {

        numerator += value;

        denominator \*= value;

        sum += numerator / denominator;

        value += 2

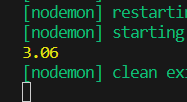
    }

    return Number(sum.toFixed(2));

}

console.log(calculateSum(10));

Результат:



1. **Вычислить:**

Код:

function calculate(): number {

    let sum = 1;

    for (let i = 1; i <= 52; i ++) {

        sum \*= Math.pow(i, 2) / (i + Math.pow(3, 2))

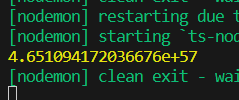
    }

    return sum;

}

console.log(calculate());

Результат:

****

**Занятие №7**

1. **Дан массив натуральных чисел. Найти сумму элементов, кратных заданному числу К.**

Код:

const numbers: number[] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15];

function findSumOfMultiples(array: number[], K: number): number {

    let sum = 0;

    for (const num of array) {

        if (num % K === 0) {

            sum += num;

        }

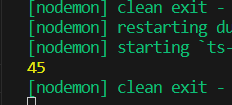
    }

    return sum;

}

console.log(findSumOfMultiples(numbers, 3));

Результат:



1. **Каждый солнечный день улитка, сидящая на дереве, поднимается вверх на 2 см, а каждый пасмурный день опускается на 1 см. В начале наблюдения улитка находилась в А см от земли на В-метровом дереве. Имеется 30-элементный массив, содержащий сведения о том, был ли соответствующий день наблюдения пасмурным или солнечным. Определите местонахождение улитки к концу 30-го дня наблюдения.**

Код:

const weatherData: boolean[] = createWeatherData(30);

const initialSnailHeight: number = 10;

const treeHeight: number = 500;

function createWeatherData(days: number): boolean[] {

    let arrData: boolean[] = []

    for (let i: number = 0; i < days; i++) {

        const value = Math.floor(Math.random() \* 100)

        if (value % 2 === 0) {

            arrData[i] = true // true - Солнечный день,

        } else {

            arrData[i] = false // false - Пасмурный день

        }

    }

    return arrData

}

function calculateSnailLocation(

days: boolean[],

initialHeight: number,

treeHeight: number

): number {

    let currentHeight = initialHeight;

    for (let i = 0; i < days.length; i++) {

        if (days[i]) {

            currentHeight += 2;

        } else {

            currentHeight -= 1;

        }

        if (currentHeight <= 0) {

            return currentHeight = 0 // Проверка, чтобы улитка не ушла ниже земли

        }

        if (currentHeight > treeHeight) {

            return currentHeight = treeHeight; // Проверка, чтобы улитка не поднялась выше дерева

        }

    }

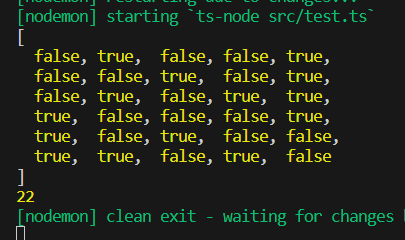
    return currentHeight;

}

console.log(weatherData)

console.log(calculateSnailLocation(weatherData, initialSnailHeight, treeHeight));

Результат:



1. **Составить программу вывода на экран любого элемента массива.**

Код:

const myArray = [1, 2, 3, 4, 5, 6, 7];

function displayArrayElement(array: any[], index: number): void {

    if (index >= 0 && index < array.length) {

        console.log(`Элемент с индексом ${index}: ${array[index]}`);

    } else {

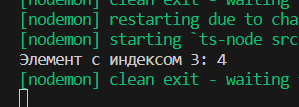
        console.log(`Индекс ${index} выходит за пределы массива.`);

    }

}

displayArrayElement(myArray, 3);

Результат:



1. **Дан вектор {zi ,}, i = 1, ..., 50. Вычислить длину этого вектора:**

Код:

function calculateVectorLength(): number {

    let sum: number = 0;

    for (let z\_i = 1; z\_i <= 50; z\_i++) {

        sum += Math.pow(z\_i, 2)

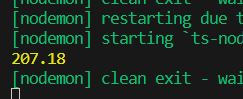
    }

    return Number((Math.sqrt(sum)).toFixed(2))

}

console.log(calculateVectorLength())

Результат:



1. **Для массива {хi ,}, i = 1,..., 20, подсчитать количество компонент, значения которых лежат в интервале [0, 1].**

Код:

const arrX = createArrayValues(20);

function createArrayValues(count: number): number[] {

    const arr: number[] = [];

    for (let i = 0; i < count; i++) {

        arr[i] = Number((Math.random() + Math.random()).toFixed(1))

    }

    return arr;

}

function countValuesInInterval(

arr: number[],

lowerBound: number,

upperBound: number

): number {

    let count = 0;

    for (let i = 0; i < arr.length; i++) {

        if (arr[i] >= lowerBound && arr[i] <= upperBound) {

            count++;

        }

    }

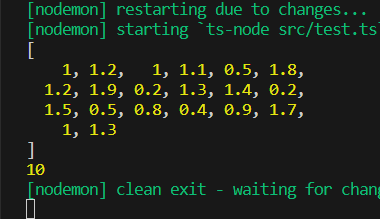
    return count;

}

console.log(arrX)

console.log(countValuesInInterval(arrX, 0, 1));

Результат:



1. **Даны два вектора {хi}, {уi}, i = 1, ..., 10, упорядоченные по возрастанию. Слить их в один вектор {zi}, i = 1, ..., 20, так чтобы сохранилась упорядоченность.**

Код:

const x = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19];

const y = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20];

function mergeSortedVectors(x: number[], y: number[]): number[] {

    let mergedVector: number[] = [];

    let i = 0;

    let j = 0;

    while (i < x.length && j < y.length) {

        if (x[i] < y[j]) {

            mergedVector.push(x[i]);

            i++;

        } else {

            mergedVector.push(y[j]);

            j++;

        }

    }

    while (i < x.length) {

        mergedVector.push(x[i]);

        i++;

    }

    while (j < y.length) {

        mergedVector.push(y[j]);

        j++;

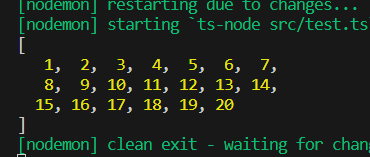
    }

    return mergedVector;

}

console.log(mergeSortedVectors(x, y));

Результат:



1. **Дан массив, состоящий из 100 целых чисел. Вывести все числа, которые встречаются в этом массиве:**

**а) несколько раз;**

**б) только по одному разу.**

Код:

const numbers = createArrayNumbers(100);

function createArrayNumbers(count: number): number[] {

    const arr: number[] = [];

    for (let i = 0; i < count; i++) {

        arr.push(Math.floor(Math.random() \* 100))

    }

    return arr

}

function findDuplicatesAndUniqueNumbers(numbers: number[]): [ number[], number[] ] {

    const numberCount = {};

    const duplicates = [];

    const unique = [];

    // Подсчитываем количество встречающихся чисел

    for (const num of numbers) {

        if (numberCount[num] === undefined) {

            numberCount[num] = 1;

        } else {

            numberCount[num]++;

        }

    }

    // Находим числа, встречающиеся несколько раз и только по одному разу

    for (const num in numberCount) {

        if (numberCount[num] > 1) {

            duplicates.push(Number(num));

        } else {

            unique.push(Number(num));

        }

    }

    return [ duplicates, unique ];

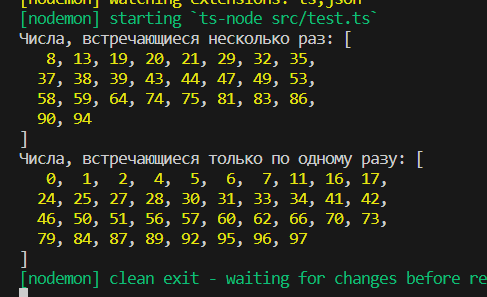
}

const [ duplicates, unique ] = findDuplicatesAndUniqueNumbers(numbers);

console.log("Числа, встречающиеся несколько раз:", duplicates);

console.log("Числа, встречающиеся только по одному разу:", unique);

Результат:



1. **В целочисленном массиве размером 10 найти значение и индекс максимального элемента.**

Код:

const array = [5, 10, 2, 8, 15, 7, 20, 3, 12, 1];

function findMaxValueAndIndex(arr: number[]): {

value: number,

index: number

} {

    let maxValue = arr[0];

    let maxIndex = 0;

    for (let i = 1; i < arr.length; i++) {

      if (arr[i] > maxValue) {

        maxValue = arr[i];

        maxIndex = i;

      }

    }

    return { value: maxValue, index: maxIndex };

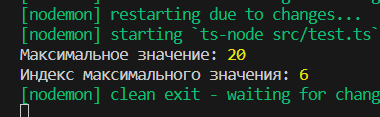
}

const { value, index } = findMaxValueAndIndex(array);

console.log(`Максимальное значение:`, value);

console.log(`Индекс максимального значения:`, index);

Результат:



1. **В массиве хранятся сведения о количестве осадков, выпавших за каждый день января. Определить общее количество осадков за январь.**

Код:

const januaryPrecipitation = [10, 5, 8, 12, 3, 15, 7, 6, 9, 8, 5, 4, 12, 10, 8, 6, 3, 14, 7, 8, 11, 13, 9, 6, 2, 4, 5, 6, 15, 8, 10];

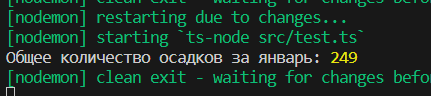
function totalPrecipitation(precipitationData: number[]): number {

    return precipitationData.reduce((acc, value) => acc + value, 0);

}

console.log('Общее количество осадков за январь:', totalPrecipitation(januaryPrecipitation));

Результат:



1. **Определить:**

**а) сумму всех элементов массива;**

**б) произведение всех элементов массива;**

**в) суму квадратов всех элементов массива;**

Код:

const myArray = [2, 4, 6, 8, 10];

function sumArrayElements(arr: number[]): number {

    return arr.reduce((acc, value) => acc + value, 0);

}

  function productOfArrayElements(arr: number[]): number {

    return arr.reduce((acc, value) => acc \* value, 1);

}

  function sumOfSquares(arr: number[]): number {

    return arr.reduce((acc, value) => acc + value \* value, 0);

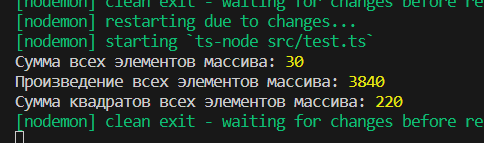
}

console.log(`Сумма всех элементов массива:`, sumArrayElements(myArray));

console.log(`Произведение всех элементов массива:`, productOfArrayElements(myArray));

console.log(`Сумма квадратов всех элементов массива:`, sumOfSquares(myArray));

Результат:



1. **Дан массив. Напечатайте:**

**а) все неотрицательные элементы;**

**б) все элементы, не превышающие числа 100.**

Код:

const myArray = [2, -5, 67, 101, 8, 42, -10, 99, 239, -321];

function printNonNegativeElements(arr: number[]): number[] {

    return arr.filter((element) => element >= 0);

}

function printElementsNotExceeding100(arr: number[]): number[] {

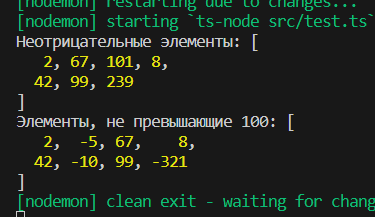
    return arr.filter((element) => element <= 100);

}

console.log("Неотрицательные элементы:", printNonNegativeElements(myArray));

console.log("Элементы, не превышающие 100:", printElementsNotExceeding100(myArray));

Результат:



1. **Дан массив целых чисел. Напечатать все четные элементы.**

Код:

const integerArray = [2, 5, 8, 11, 14, 17, 20, 23];

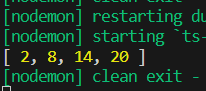
function printEvenElements(arr: number[]): number[] {

    return arr.filter((element) => element % 2 === 0);

}

console.log(printEvenElements(integerArray));

Результат:



1. **Найти сумму элементов массива А[1..12] действительных чисел, имеющих четные индексы.**

Код:

const array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12];

function sumElementsWithEvenIndices(arr: number[]): number {

    let sum = 0;

    for (let i = 0; i < arr.length; i += 2) {

      sum += arr[i];

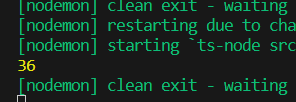
    }

    return sum;

  }

  console.log(sumElementsWithEvenIndices(array));

Результат:



1. **Найти минимальный и максимальный элементы массива М[1..10] целых чисел и поменять их местами.**

Код:

const array = [5, 40, 2, 8, 15, 7, 20, 3, 12, 1];

function findMinMaxAndSwap(arr: number[]): void {

    let minIndex = 0;

    let maxIndex = 0;

    for (let i = 1; i < arr.length; i++) {

      if (arr[i] < arr[minIndex]) {

        minIndex = i;

      }

      if (arr[i] > arr[maxIndex]) {

        maxIndex = i;

      }

    }

    const minValue = arr[minIndex];

    const maxValue = arr[maxIndex];

    arr[minIndex] = maxValue;

    arr[maxIndex] = minValue;

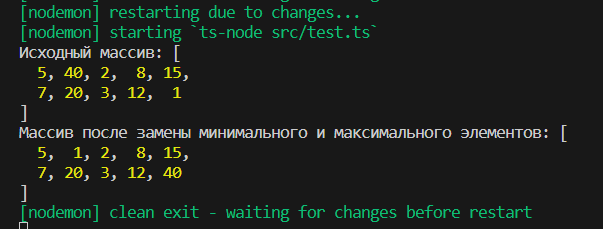
  }

  console.log("Исходный массив:", array);

  findMinMaxAndSwap(array);

  console.log("Массив после замены минимального и максимального элементов:", array);

Результат:



1. **Найти количество элементов массива S[1..15] целых чисел кратных 5.**

Код:

 const array = [10, 5, 7, 15, 20, 8, 25, 3, 5, 35, 12, 1, 50, 18, 9];

 function countMultiplesOf5(arr: number[]): number {

    let count = 0;

    for (let i = 0; i < arr.length; i++) {

      if (arr[i] % 5 === 0) {

        count++;

      }

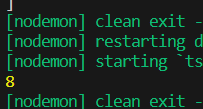
    }

    return count;

  }

  console.log(countMultiplesOf5(array));

Результат:



1. **Найти сумму элементов массива М[1..10] действительных чисел до первого отрицательного элемента.**

Код:

 const arrNum: number[] = [2, 4, 6, -3, 8, 10, -5, 12, 14, 16];

 function findSumBeforeNegativeElement(arr: number[]): number {

  let sum = 0;

  for (let i = 0; i < arr.length; i++) {

    if (arr[i] < 0) {

      break;

    }

    sum += arr[i];

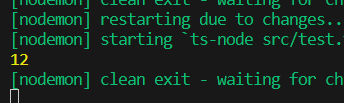
  }

   return sum;

 }

 console.log(findSumBeforeNegativeElement(arrNum));

Результат:



1. **В целочисленной последовательности есть нулевые элементы. Создать массив из номеров этих элементов.**

Код:

 const sequence: number[] = [1, 0, 0, 5, 0, 3, 0, 0, 8];

 function findZeroIndices(arr: number[]): number[] {

  const zeroIndices: number[] = [];

  for (let i = 0; i < arr.length; i++) {

    if (arr[i] === 0) {

      zeroIndices.push(i);

    }

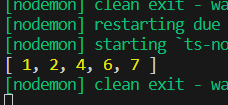
  }

   return zeroIndices;

 }

 console.log(findZeroIndices(sequence));

Результат:



1. **Дана последовательность целых чисел а1, а2,..., аn. Выяснить, какое число встречается раньше – положительное или отрицательное?**

Код:

 const sequence: number[] = [3, -5, 0, 7, -2, 8];

 function findFirstPositiveOrNegative(arr: number[]): string {

  for (let i = 0; i < arr.length; i++) {

    if (arr[i] > 0) {

      return "Первым встречается положительное число";

    }

    if (arr[i] < 0) {

      return "Первым встречается отрицательное число";

    }

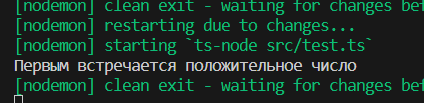
    return "Первым встречается ноль";

  }

 }

 console.log(findFirstPositiveOrNegative(sequence));

Результат:



1. **В массиве, состоящем из оценок, двойки заменить на тройки.**

Код:

const gradesArray: number[] = [4, 2, 5, 3, 2, 4, 3];

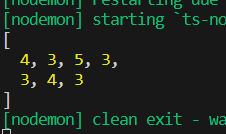
 function replaceTwosWithThrees(grades: number[]): number[] {

    return grades.map((grade) => (grade === 2 ? 3 : grade));

 }

console.log(replaceTwosWithThrees(gradesArray));

Результат:



1. **Дана последовательность целых чисел а1, а2,..., аn. Вывести на экран только те числа, для которых выполняется неравенство аi<= i.**

Код:

 const sequence: number[] = [3, 1, 5, 2, 7, 4, 6];

 function filterAndPrintNumbers(arr: number[]): void {

   for (let i = 0; i < arr.length; i++) {

     if (arr[i] <= i + 1) {

       console.log(arr[i]);

     }

   }

 }

 filterAndPrintNumbers(sequence);

Результат:

