

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных.

по лабораторной работе №2

Дисциплина: Языки программирования для работы с большими данными.

(И.О. Фамилия)

(И.О. Фамилия)

Москва, 2022

Лабораторная работа №2

Задание: Вариант 1

3. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.

Ход работы: Код программы

```
import java.util.Scanner;

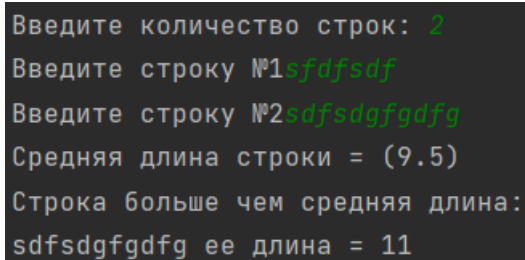
public class Program1 {
    public static void main(String[] args) {
        int n = 0;
        System.out.print("Введите количество строк: ");
        Scanner scanner1 = new Scanner(System.in);
        n = scanner1.nextInt();
        String[] strings = new String[n];
        Scanner scanner2 = new Scanner(System.in);

        for(int i = 0; i < strings.length; i++){
            System.out.print("Введите строку №" + (i+1) );
            strings[i] = scanner2.nextLine();
        }

        float length = 0f;
        for(String item : strings){
            length += (float) item.length();
        }

        length /= strings.length;

        System.out.println("Средняя длина строки = (" + length + ")");
        for(int i = 0; i < n; i++){
            if (strings[i].length() > length){
                System.out.println("Строка больше чем средняя длина: ");
                System.out.println(strings[i] + " ее длина = " + strings[i].length());
            }
        }
    }
}
```



```
Введите количество строк: 2
Введите строку №1sfdfsdf
Введите строку №2sdfsdfgfdgfdg
Средняя длина строки = (9.5)
Строка больше чем средняя длина:
sdfsdfgfdgfdg ее длина = 11
```

Рисунок 1. Результат работы программы

Задание: Вариант 1

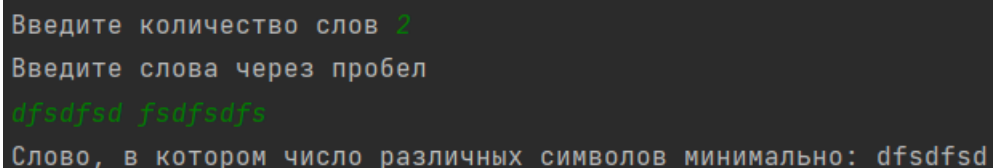
4. Ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

Ход работы:

Код программы

```
import java.util.*;
public class Program2 {
    public static void main(String[] args) {
        int n; //количество строк
        String[] words;
        String text;
        System.out.print("Введите количество слов ");
        Scanner scan = new Scanner(System.in);
        if (scan.hasNextInt()) {
            System.out.println("Введите слова через пробел");
            n = Integer.parseInt(scan.nextLine());
            text = scan.nextLine();
            words = text.split(" ", n);
            String wordMinCharVar = words[0];
            int sizeMinCharVar = new HashSet<String>(Arrays.asList(words[0].split(""))).size();

            for (String word : words) {
                Set<String> set = new HashSet<String> (Arrays.asList (word.split ("")));
                if (set.size () < sizeMinCharVar) {
                    wordMinCharVar = word;
                    sizeMinCharVar = set.size ();
                }
            }
            System.out.println ("Слово, в котором число различных символов минимально: " +
wordMinCharVar);
        }
    }
}
```



```
Введите количество слов 2
Введите слова через пробел
dfsd fsd fsd
Слово, в котором число различных символов минимально: dfsd
```

Рисунок 2. Результат работы программы

Задание: Вариант 2. Ввести с консоли n – размерность матрицы $a[n][n]$.
Задать значения элементов матрицы в интервале значений от $-n$ до n с помощью датчика случайных чисел.

3. Найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.

Ход работы: Код программы

```
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;
public class test3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите размер матрицы");
        int size = sc.nextInt();
        int arr[][] = new int[size][size];
        fillRandom(arr, size); //заполнение случ
        printArr(arr);
        System.out.println();
        inDecrease(arr); //макс последовательность возрастающих/убыв элементов
    }
    public static void fillRandom(int a[], int d) {
        Random random = new Random();
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a.length; j++) {
                a[j][i] = random.nextInt(2 * d + 1) - d;
            }
        }
    }

    public static void printArr(int[][] a) {

        for(int[] x:a){
            for(int z:x){
                System.out.print(z+" ");
            } System.out.println();
        }

    }

    public static void inDecrease(int[][] b) {
        int[] f = new int[b.length * b.length]; //преобразую в одномерный массив
        int c = 0; int r=0; int max=0; int h=0; int m=0; int min=0; int h1=0;
        for(int[] x:b){
            for(int z:x){
```

```

        f[c++] = z;
    }
}
for (int g=0;g<f.length-1;g++){
    if(f[g]<f[g+1]){//если последующий элемент больше
        r+=1;
        if(r>max){max=r;h=g+1;}//запоминаем количество возрастных и
индекс последнего возрастающего элем
    }else{r=0;}//возрастание прервалась
    if(f[g]>f[g+1]){
        m+=1;
        if(m>min){min=m;h1=g+1;}//запоминаем количество возрастных и
индекс последнего убыв элем
    } else{m=0;}// убывание прервалась
    }
    System.out.println("Максимальная последовательность возрастающих
чисел:");
    int[] l=new int[max+1];
    System.arraycopy(f,(h-max),l,0,(max+1));//копирую последовательность
возр элем в новый массив
    System.out.println(Arrays.toString(l)+" "+(max+1));
    System.out.println("Максимальная последовательность убывающих
чисел:");
    int[] t=new int[min+1];
    System.arraycopy(f,(h1-min),t,0,(min+1));//копирую последовательность
убыв элем в новый массив
    System.out.println(Arrays.toString(t)+" "+(min+1));
}
}

```

```

Введите размер матрицы
3
-1  2  1
2   0 -1
-3  2  1

Максимальная последовательность возрастающих чисел:
[-1, 2] 2
Максимальная последовательность убывающих чисел:
[2, 0, -1, -3] 4

```

Рисунок 3. Результат работы программы

Задание: Вариант 2. Ввести с консоли n – размерность матрицы $a[n][n]$.
Задать значения элементов матрицы в интервале значений от $-n$ до n с помощью датчика случайных чисел.

4. Найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки.

Ход работы: Код программы

```
import java.util.*;
public class test {
    public static void main(String[] args) {
        System.out.print("Введите диапазон n: ");
        Scanner scan = new Scanner(System.in);
        int n = Integer.parseInt(scan.nextLine());
        int min = n;
        int max = -n;
        int[][] matrix = new int[n][n]; //создание массива a размером 10*10

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = (int) (Math.random() * (max - min)) + min; //генерируем
число от -n до n.
            }
        }

        for (int i = 0; i < n; i++) { //вывод матрицы
            for (int j = 0; j < n; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.print("\n"); //переход на новую строку
        }
        int sum = 0; // сумма элементов между первыми двумя положительными
числами в строке
        int positive_cnt = 0; // счетчик положительных чисел локальный для
каждой строки
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {

                if (matrix[i][j] >= 0) {
                    positive_cnt++;
                }
            }

            //если количество положительных чисел в строке больше или равно
двум,
```

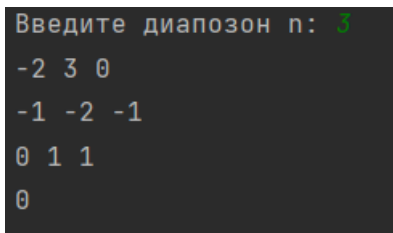
```

        //то это удовлетворяет нашему условию для нахождения суммы между
ними
        if (positive_cnt >= 2) {
            positive_cnt = 0; //обнуляем счетчик положительных чисел

            //снова проходим по строке циклом

            for (int j = 0; j < n - 1; j++) {
                //находим положительное число и увеличиваем значения
счетчика
                if (matrix[i][j] >= 0) {
                    positive_cnt++;
                }
                //если значение счетчика равно 1 и последующий элемент строки меньше 0,
                //то прибавляем следующий элемент строки к сумме, пока значение
счетчика не станет равным 2
                if (positive_cnt == 1 && matrix[i][j + 1] < 0) {
                    sum += matrix[i][j + 1];
                }
            }
        }
        System.out.println(sum);
    }
}

```



```

Введите диапазон n: 3
-2 3 0
-1 -2 -1
0 1 1
0

```

Рисунок 4. Результат работы программы

Вывод: лабораторная работа была выполнена в соответствие с заданием и полученные верные результаты работ программ