Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра “Математической кибернетики и информационных технологий”

дисциплина «Структуры и алгоритмы обработки данных»

Курсовая работа

“Реализация поставленных задач”

Выполнил: студенты группы БФИ1901

Козырев Сергей Владимирович

Проверил:

Кутейников Иван Алексеевич

Москва 2021

Оглавление

1.Задание на Курсовую Работу.

Задание на лабораторную работу показано на рисунках 1-8.

Задача 1. «Треугольник с максимальным периметром»

Массив `A` состоит из целых положительных чисел - длин отрезков. Составьте из трех отрезков такой треугольник, чтобы его периметр был максимально возможным. Если невозможно составить треугольник с положительной площадью - функция возвращает `0`.

Пример 1.1:
Ввод: `[2,1,2]`
Вывод: `5`

Пример 1.2:
Ввод: `[1,2,1]`
Вывод: `0`

Пример 1.3:
Ввод: `[3,2,3,4]`
Вывод: `10`

Пример 1.4:
Ввод: `[3,6,2,3]`
Вывод: `8`

Ограничения:

- $3 \leqslant len(A) \leqslant 10000$
- $1 \leqslant A[i] \leqslant 10^6$

Рисунок 1 – Задание на работу.

Задача 2. «Максимальное число»

Дан массив неотрицательных целых чисел `nums`. Расположите их в таком порядке, чтобы вместе они образовали максимально возможное число.

Замечание: Результат может быть очень большим числом, поэтому представьте его как `string`, а не `integer`.

Пример 2.1:
Ввод: `nums = [10,2]`
Вывод: `"210"`

Пример 2.2:
Ввод: `nums = [3,30,34,5,9]`
Вывод: `"9534330"`

Пример 2.3:
Ввод: `nums = [1]`
Вывод: `"1"`

Пример 2.4:
Ввод: `nums = [10]`
Вывод: `"10"`

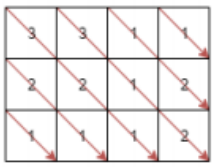
Ограничения:

- $1 \leqslant len(nums) \leqslant 100$
- $0 \leqslant nums[i] \leqslant 10^9$

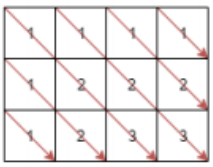
Рисунок 2 – Задание на работу.

Задача 3. «Сортировка диагоналей в матрице»

Дана матрица `mat` размером `m * n`, значения - целочисленные. Напишите функцию, сортирующую каждую диагональ матрицы по возрастанию и возвращающую получившуюся матрицу.



Before



After

Пример 3.1:

Ввод: `mat = [[3, 3, 1, 1], [2, 2, 1, 2], [1, 1, 1, 2]]`

Вывод: `[[1, 1, 1, 1], [1, 2, 2, 2], [1, 2, 3, 3]]`

Пример 3.2:

Ввод: `mat = [[11, 25, 66, 1, 69, 7], [23, 55, 17, 45, 15, 52], [75, 31, 36, 44, 58, 8], [22, 27, 33, 25, 68, 4], [84, 28, 14, 11, 5, 50]]`

Вывод: `[[5, 17, 4, 1, 52, 7], [11, 11, 25, 45, 8, 69], [14, 23, 25, 44, 58, 15], [22, 27, 31, 36, 50, 66], [84, 28, 75, 33, 55, 68]]`

Ограничения:

- $m == len(mat)$
- $n == len(mat[i])$
- $1 \leqslant m, n \leqslant 100$
- $1 \leqslant mat[i][j] \leqslant 100$

Рисунок 3 – Задание на работу.

Задача 1. «Шарики и стрелы»

Некоторые сферические шарики распределены по двухмерному пространству. Для каждого шарика даны x-координаты начала и конца его горизонтального диаметра. Так как пространство двумерно, то y-координаты не имеют значения в данной задаче. Координата x_{start} всегда меньше x_{end}.

Стрелу можно выстрелить строго вертикально (вдоль y-оси) из разных точек x-оси. Шарик с координатами x_{start} и x_{end} уничтожается стрелой, если она была выпущена из такой позиции x, что $x_{start} \leqslant x \leqslant x_{end}$. Когда стрела выпущена, она летит в пространстве бесконечное время (уничтожая все шарики на пути).

Дан массив `points`, где `points[i] = [`x_{start}`,` x_{end}`]`. Напишите функцию, возвращающую минимальное количество стрел, которые нужно выпустить, чтобы уничтожить все шарики.

Пример 1.1:

Ввод: `points = [[10,16],[2,8],[1,6],[7,12]]`
Вывод: `2`

Пример 1.2:

Ввод: `points = [[1,2],[3,4],[5,6],[7,8]]`
Вывод: `4`

Пример 1.3:

Ввод: `points = [[1,2],[2,3],[3,4],[4,5]]`
Вывод: `2`

Пример 1.4:

Ввод: `points = [[1,2]]`
Вывод: `1`

Пример 1.5:

Ввод: `points = [[2,3],[2,3]]`
Вывод: `1`

Ограничения:

- $0 \leqslant len(points) \leqslant 10^4$
- $len(points[i]) == 2$
- $-2^{31} \leqslant x_{start} < x_{end} \leqslant 2^{31} - 1$

Рисунок 4 – Задание на работу.

ЗАДАЧА 1

Даны две строки: s1 и s2 с одинаковым размером, проверьте, может ли некоторая перестановка строки s1 “победить” некоторую перестановку строки s2 или наоборот.

Строка x может “победить” строку y (обе имеют размер n), если x[i]> = y [i] (в алфавитном порядке) для всех i от 0 до n-1.

Рисунок 5 – Задание на работу.

ЗАДАЧА 2

Дана строка s, вернуть самую длинную полиндромную подстроку в s.

Примеры:

```
Input: s = "babad"
Output: "bab"
Note: "aba" is also a valid answer.

Input: s = "cbbd"
Output: "bb"
```

Рисунок 6 – Задание на работу.

ЗАДАЧА 3

Вернуть количество отдельных непустых подстрок текста, которые могут быть записаны как конкатенация некоторой строки с самой собой (т.е. она может быть записана, как a + a, где a - некоторая строка).

Примеры:

```
Input: text = "abcabcabc"

Output: 3

Explanation: The 3 substrings are "abcabc", "bcabca"
and "cabcab".
```

Рисунок 7 – Задание на работу.

Задача 1. «Стопки монет»

На столе стоят $3n$ стопок монет. Вы и ваши друзья Алиса и Боб забираете стопки монет по следующему алгоритму:

1. Вы выбираете 3 стопки монет из оставшихся на столе.
2. Алиса забирает себе стопку с максимальным количеством монет.
3. Вы забираете одну из двух оставшихся стопок.
4. Боб забирает последнюю стопку.
5. Если еще остались стопки, то действия повторяются с первого шага.

Дан массив целых положительных чисел `piles`. Напишите функцию, возвращающую максимальное число монет, которое вы можете получить.

Пример 1.1:

Ввод: `piles = [2,4,1,2,7,8]`
Вывод: `9`

Пример 1.2:

Ввод: `piles = [2,4,5]`
Вывод: `4`

Пример 1.3:

Ввод: `piles = [9,8,7,6,5,1,2,3,4]`
Вывод: `18`

Ограничения:

- $3 \leqslant len(piles) \leqslant 10^5$
- $len(piles) \, mod \, 3 == 0$
- $1 \leqslant piles[i] \leqslant 10^4$

Рисунок 8 – Задание на работу.

2.Ход работы.

Для начала распределим задания по блокам. Первые три задачи в первом блоке с названием (Zadachi1), далее отдельная задача «Шарики и стрелы» в отдельном блоке (Zadachi2), после чего в отдельном блоке опять 3 задачи (Zadachi3) и наконец отдельная задача «Стопки монет» в блоке (Zadachi4). Приступим к реализации заданий по блокам.

3.Листинг программы (Zadachi)

2.1 Листинг программы Zadachi1.

```
package Zadachi;
import java.util.Arrays;
import java.util.Scanner;
import java.util.Stack;

public class Zadachi1 {
    //Задачи из первого файла

        public static void main(String[] args) {
            Zadachi1 s = new Zadachi1();
            System.out.println("Задание 1:");
            s.Z1();
            System.out.println("Задание 2:");
            Scanner scanner = new Scanner(System.in);
            System.out.println("Введите количество чисел массива nums");
            int z=scanner.nextInt();
            int[] int_nums = new int[z];
            for (int i=0;i<z;i++){
                int_nums[i] = scanner.nextInt();
            }
            System.out.println(Z2(int_nums));
            System.out.println("Задание 3:");
            Scanner sc = new Scanner(System.in);
            System.out.println("Введите размерность матрицы:");
            String m1 = sc.nextLine();
            String n1 = m1;
            System.out.println("Введите минимальный элемент матрицы:");
            String min_lim1 = sc.nextLine();
            System.out.println("Введите максимальный элемент матрицы:");
            String max_lim1 = sc.nextLine();
            if (n1.equals(""))
                n1 = "50";
            if (m1.equals(""))
```

```
                m1 = "50";
            if (min_lim1.equals(""))
                min_lim1 = "-250";
            if (max_lim1.equals(""))
                max_lim1 = "1013";
            int n = Integer.parseInt(n1);
            int m = Integer.parseInt(m1);
            int min_lim = Integer.parseInt(min_lim1);
            int max_lim = Integer.parseInt(max_lim1);
            int[][] arr = new int[n][m];
            for (int i = 0; i < n; i++) {
                System.out.print("\n");
                for (int j = 0; j < m; j++) {
                    arr[i][j] = (int) ((Math.random() * (max_lim - min_lim)) + mi
n_lim);
                    System.out.print(arr[i][j] + "\t");
                }
                System.out.println();
            }
            Zadachi1 f = new Zadachi1();
            int[][] rez = f.Z3(arr,n,m);
            System.out.println("----------------------------------------------
----------------------");
            for (int i=0;i<n;i++){
                System.out.print("\n");
                for (int j=0;j<n;j++){
                    System.out.print(rez[i][j]+"\t");
                }
                System.out.println();
            }
        }
        public void Z1() {
            System.out.println("Введите размер массива:");
            Scanner sc = new Scanner(System.in);
            int n = sc.nextInt();
            int[] arr = new int[n];
            for (int i = 0; i < n; i++) {
                arr[i] = (int) (Math.random() * 100);
            }
            System.out.println("Сгенерированный массив:");
            for (int i = 0; i < n; i++) {
                System.out.print(arr[i] + " ");
            }
            System.out.print("\n");
            maxPerimeter(arr, n);
        }
        static void maxPerimeter (int[] arr, int n) {
            int maxi = 0;
// инициализируем максимальный периметр как 0.

            for (int i = 0; i < n - 2; i++)
```

```
// подбираем 3 разных элемента
            {
// из массива
                for (int j = i + 1; j < n - 1; j++) {
                    for (int k = j + 1; k < n; k++) {
                        int a = arr[i];
                        int b = arr[j];
                        int c = arr[k];
                        if (a < b + c && b < c + a && c < a + b)
//проверяем, a, b, c образуют треуг. или нет
                        {
                            maxi = Math.max(maxi, a + b + c);
// если он образует треугольник
                        }
// затем обновляем максимум
                    }
                }
            }
            if (maxi > 0)
// Если максимальный периметр ненулевой
                System.out.println("Максимальный периметр: " + maxi);
            else
// иначе треугольник не строится
                System.out.println("Невозможно составить треугольник");
        }
        public static String Z2(int[] int_nums) {
            String str = "";
            int k = 0;
            for (int i= 0; i < int_nums.length; i++){
                for (int j = 1; j < int_nums.length-i; j++) {
                    String x = Integer.toString(int_nums[j-
1])+Integer.toString(int_nums[j]);
                    String y = Integer.toString(int_nums[j])+Integer.toString(int
_nums[j-1]);
                    if (x.compareTo(y)<0) {
                        k=int_nums[j];int_nums[j]=int_nums[j-1];int_nums[j-1]=k;
                    }
                }
            }
            for (int item:int_nums) {
                str+=Integer.toString(item);
            }
            return str;
        }
    public static int[][] Z3(int[][] a,int n,int m){
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                sort(a,i,j);
            }

        }
```

```
        return a;
    }

    public static void sort(int [][] a, int i, int j){


        if (i==0 || j==0){}else{
            if(a[i][j]<a[i-1][j-1]){
                int k = a[i][j];
                a[i][j]=a[i-1][j-1];
                a[i-1][j-1]=k;
            }
            sort(a,i-1,j-1);
        }
    }


}
```

2.2 Листинг программы Zadachi2.

```
package Zadachi;
import java.util.Arrays;
import java.util.Comparator;
import java.util.Scanner;

public class Zadachi2{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите кол-во пар чисел:");
        int xCoord = Integer.parseInt(scanner.nextLine());
// Считываем, сколько будет пар в массиве
        if (xCoord > 0) {
// Если не 0, то продолжаем
            int[][] points = new int[xCoord][2];
// Заводим массив
            for (int i = 0; i < xCoord; i++) {
// Циклом заполняем координаты
                System.out.println("Введите числа для заполнения " + (i + 1) + "-
х координат xStart и xEnd через Enter:");
                points[i][0] = Integer.parseInt(scanner.nextLine());
                points[i][1] = Integer.parseInt(scanner.nextLine());
                if (points[i][1] <= points[i][0]) {
                    System.out.println("Стартовая позиция шарика не может быть ме
ньше конечной!");
                    System.exit(0);
                }
            }
```

```
            System.out.println("Ответ: " + findMinArrowShots(points)); // Выводим
 результат
        } else
            System.out.println("Массив не может иметь 0 или отрицательное кол-
во пар!");
    }

    public static int findMinArrowShots(int[][] points) {
        Arrays.sort(points, Comparator.comparingInt(a -> a[1]));
// Сортировка по координате y
        System.out.println("Массив отсортирован: " + Arrays.deepToString(points))
;
        int arrow = 1;
// Переменная кол-ва стрел
        int end = points[0][1];
// Берём первый шарик
        for (int i = 1; i < points.length; i++) {
            if (end < points[i][0]) {
// Если шарик оказался дальше по координате xEnd
                arrow++;
// Прибавляем кол-во стрел
                end = points[i][1];
// Перемещаем конечную точку на этот шар
            }
        }
        return arrow;
    }
}
```

2.3 Листинг программы Zadachi3.

```
package Zadachi;
import java.util.*;
public class Zadachi3 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Задание 1:");
        System.out.println("Введите первую строку");
        String s1 = in.nextLine();
        System.out.println("Введите вторую строку");
        String s2 = in.nextLine();

        System.out.println("Результат: " + checkIfCanBreak(s1,s2));
        System.out.println("Задание 2:");
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите строку:");
        String str1 =scanner.nextLine();
        System.out.println("Введенная строка: " + str1);
        System.out.println("Длина самой большой строки полиндрома: " + longPalSub
str(str1));
```

```
        System.out.println("Задание 3:");
        System.out.println("Введите строку:");
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        System.out.println( distinctEchoSubstrings(s));
    }
    static void printSubStr(String str1, int l, int h) {
        System.out.println(str1.substring(l, h + 1));
    }
    static int longPalSubstr(String str1) {
        int n = str1.length();
        boolean table[][] = new boolean[n][n];
        int mLength = 1;
        for (int i = 0; i < n; ++i)
            table[i][i] = true;
        int strt = 0;
        for (int i = 0; i < n - 1; ++i) {
            if (str1.charAt(i) == str1.charAt(i + 1)) {
                table[i][i + 1] = true;
                strt = i;
                mLength = 2;
            }
        }
        for (int k = 3; k <= n; ++k) {
            for (int i = 0; i < n - k + 1; ++i) {
                int j = i + k - 1;
                if (table[i + 1][j - 1] && str1.charAt(i) == str1.charAt(j)) {
                    table[i][j] = true;
                    if (k > mLength) {
                        strt = i;
                        mLength = k;
                    }
                }
            }
        }
        System.out.print("Наибольшая строка полиндром в ввведенной строке: ");
        printSubStr(str1, strt, strt + mLength - 1);
        return mLength;
    }
    public static boolean checkIfCanBreak(String s1, String s2) {
        int []arr1=new int[26];
        int []arr2=new int[26];

        for(int i=0;i<s1.length();i++){
            arr1[s1.charAt(i)-'a']++;
        }

        for(int i=0;i<s2.length();i++){
            arr2[s2.charAt(i)-'a']++;
        }
```

```
        int count1=0;
        int count2=0;
        int greater=0;
        int smaller=0;

        for(int i=0;i<26;i++){
            count1+=arr1[i];
            count2+=arr2[i];

            if(count2>count1){
                smaller++;
            }else if(count1>count2){
                greater++;
            }
            if(smaller>0 && greater>0)
                return false;
        }
        return true;
    }
    public static int distinctEchoSubstrings(String text) {
        if (text == null || text.length() == 0) return 0;
        Set<String> stringSet = new HashSet<>();

        for (int right = 1; right <= text.length(); ++right)
        {
            for (int left = 0; left < right; ++left)
            {
                if (right - left <= 1) continue;
                String subStr = text.substring(left, right);
                if (isEchoString(subStr)) {
                    stringSet.add(subStr);
                }
            }
        }

        return stringSet.size();
    }

    private static boolean isEchoString(String subStr)
    {
        if (subStr.length() % 2 != 0) return false;

        if (subStr.substring(0, subStr.length()/2).equals(subStr.substring(subStr
.length()/2, subStr.length())))
            return true;
        return false;
    }

}
```

2.4 Листинг программы Zadachi4.

```java
package Zadachi;
import  java.util.Arrays;
import java.util.Scanner;

public class Zadachi4 {
    public static void main(String[] args) {
        Zadachi4 s = new Zadachi4();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Кол-во стопок монет");
        int z=scanner.nextInt();
        int[] piles = new int[z];
        System.out.println("Введите стопки монет");
        for (int i = 0; i < z; i++){
            piles[i] = scanner.nextInt();
        }
        s.max_coins(piles);
    }
    public void max_coins(int[] piles){
        Arrays.sort(piles);
        int sum = 0;
        int i =piles.length - 2;
        int j = 0;
        while(j++ < piles.length / 3){
            sum += piles[i];
            i -=2;
        }
        System.out.println(sum);
    }
}
```

4.Результаты работы блоков с задачами

Результат работы блока Zadachi1 показан на рисунке 9.

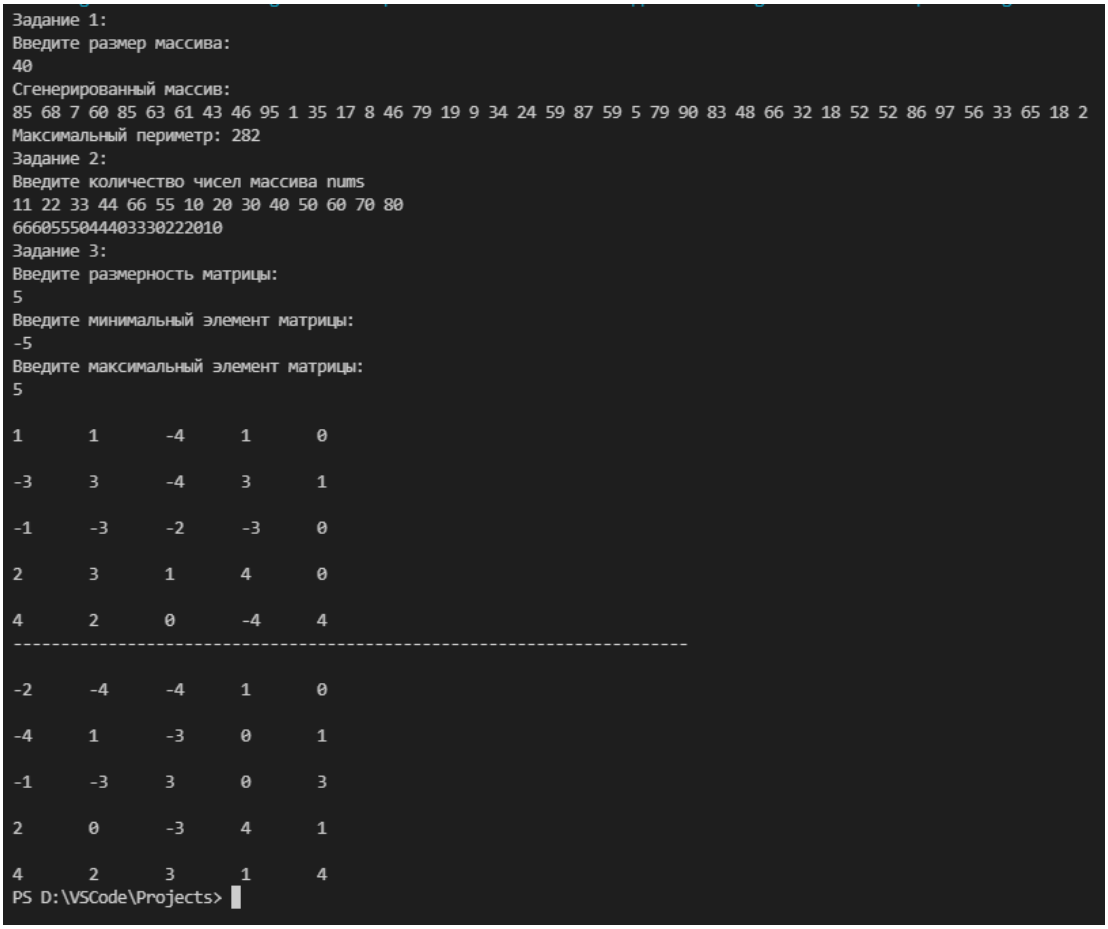


Рисунок 9 – Результат работы.

Результат работы блока Zadachi2 показан на рисунке 10.

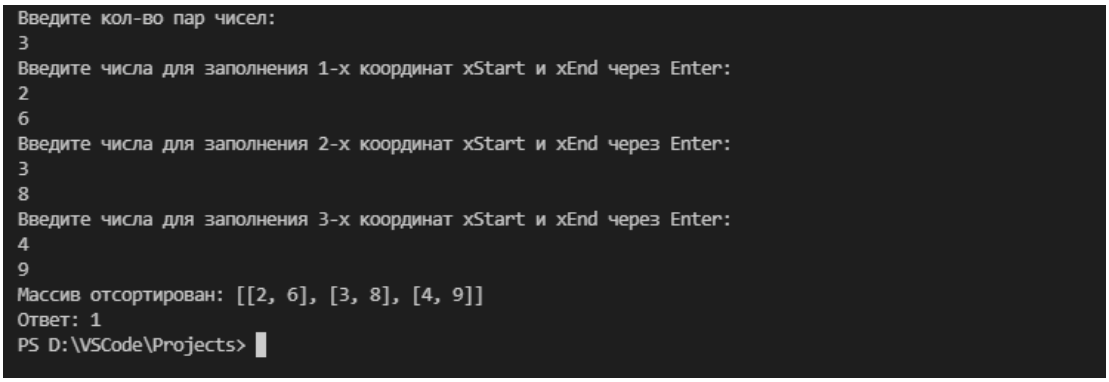


Рисунок 10 – Результат работы.

Результат работы блока Zadachi3 показан на рисунке 11.

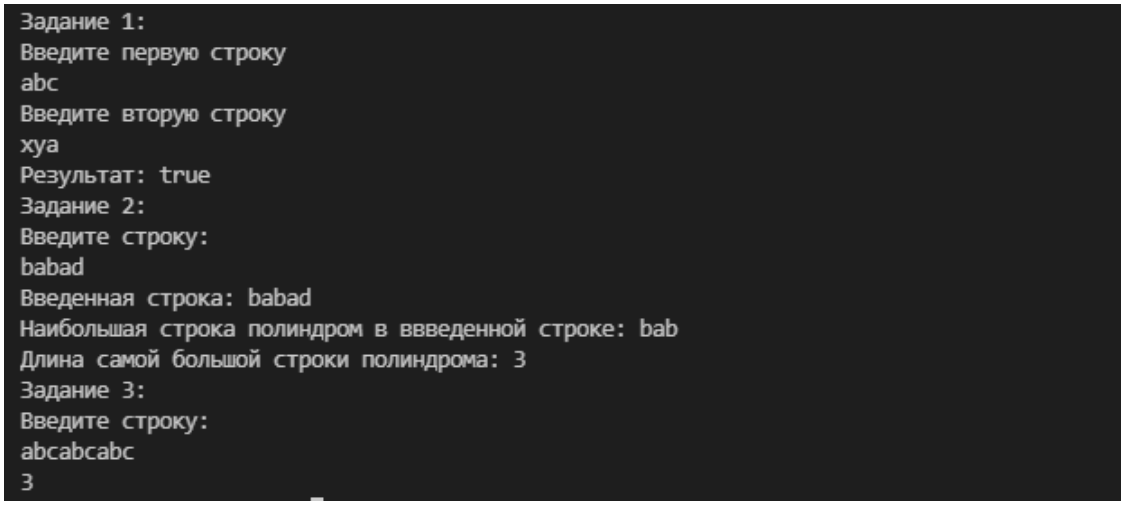


Рисунок 11 – Результат работы.

Результат работы блока Zadachi4 показан на рисунке 12.

Рисунок 12 – Результат работы.

5.Вывод

Выполнив данную курсовую работу, мы закрепили знания по курсу структур и алгоритмов обработки данных, научились реализовывать программы, использующие алгоритмы сортировок и поисков, работать с массивами данных , а также реализовывать сложные алгоритмические программы.

Список использованных источников

1 ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

2 ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления.