**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего**

**Образование**

**Ордена Трудового Красного Знамени**

**«Московский технический университет связи и информатики»**

**Кафедра «МКиИТ»**

**дисциплина «СиАОД»**

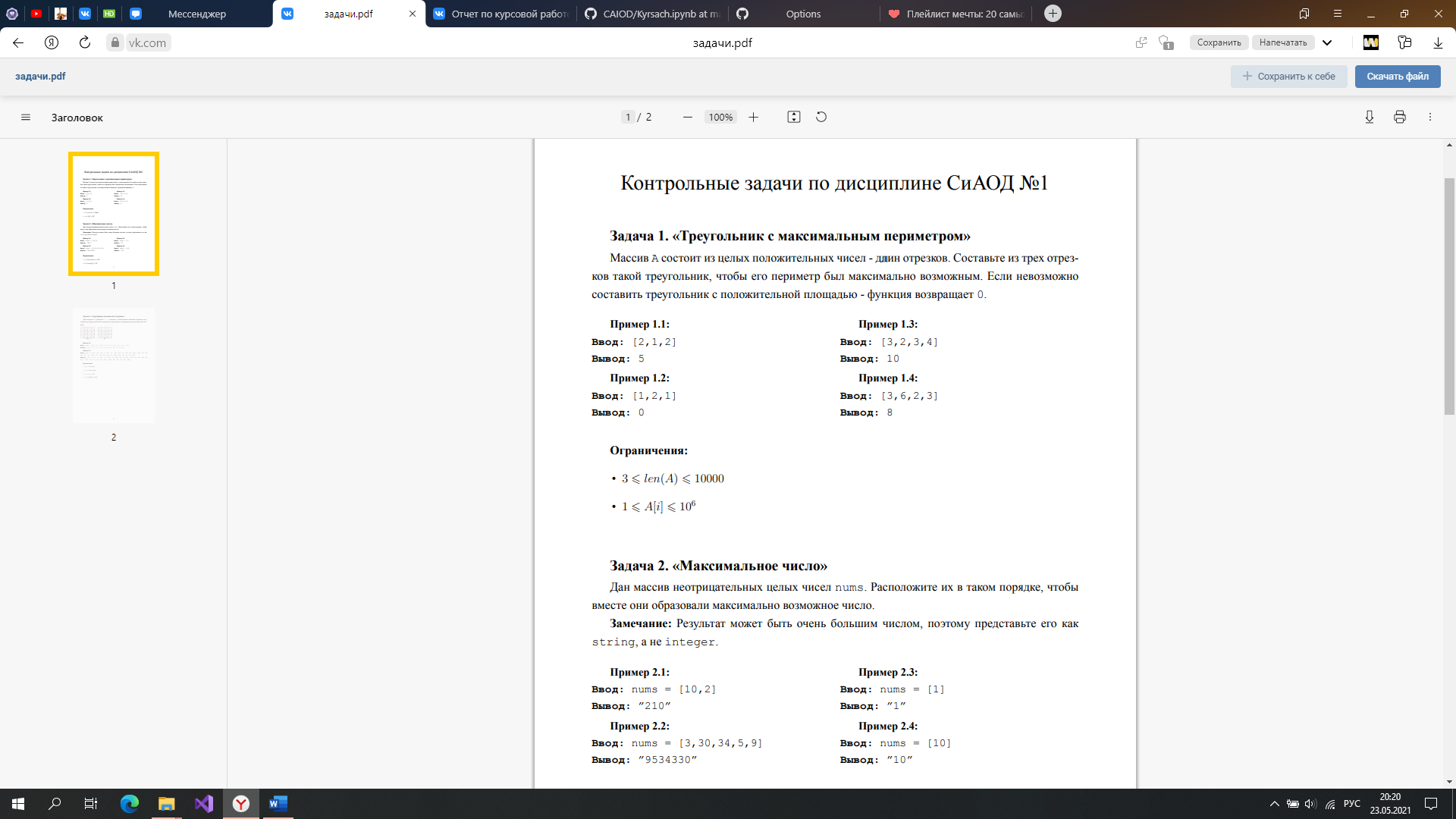
Отчет по Задачам

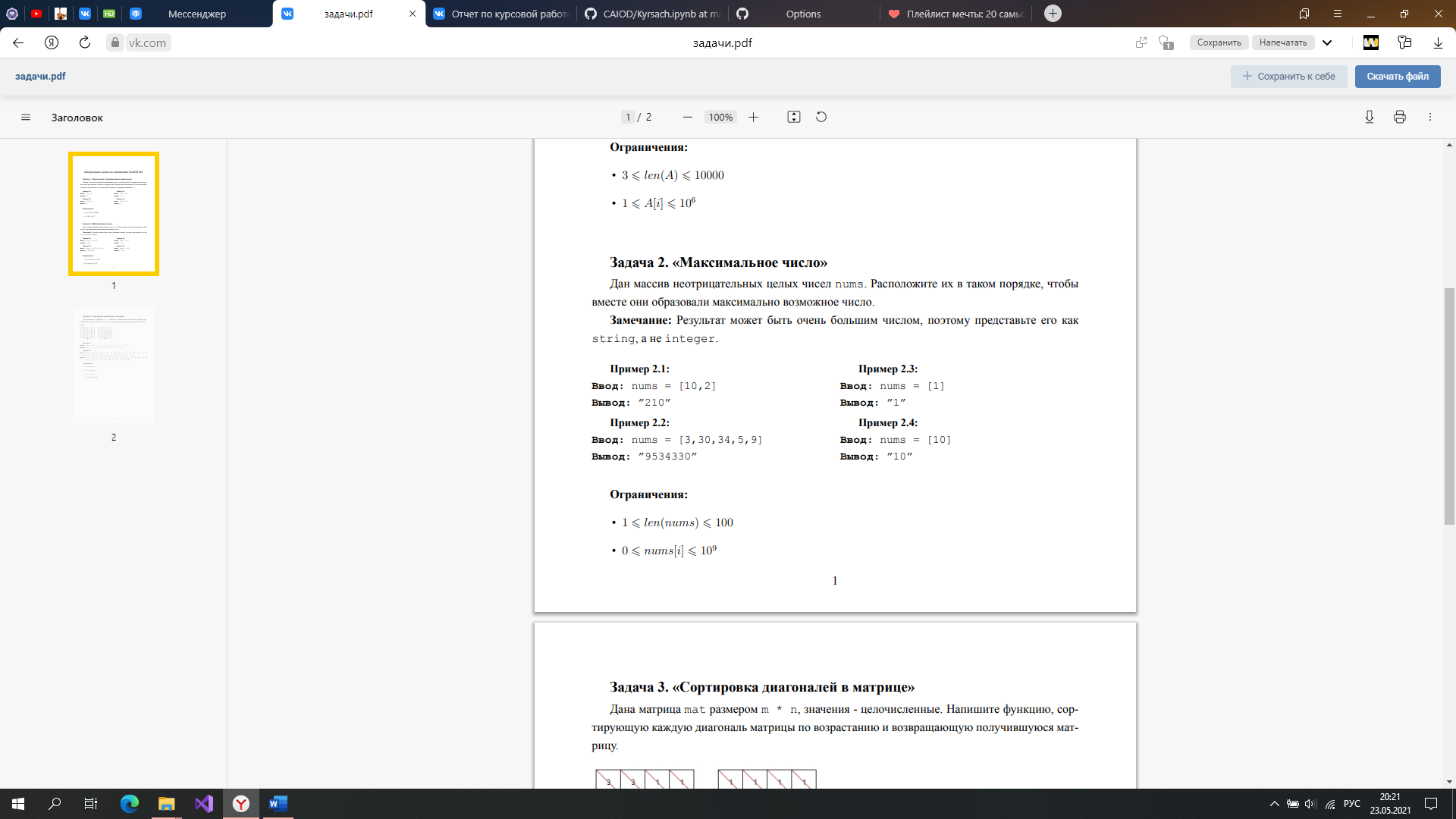
Подготовил студент

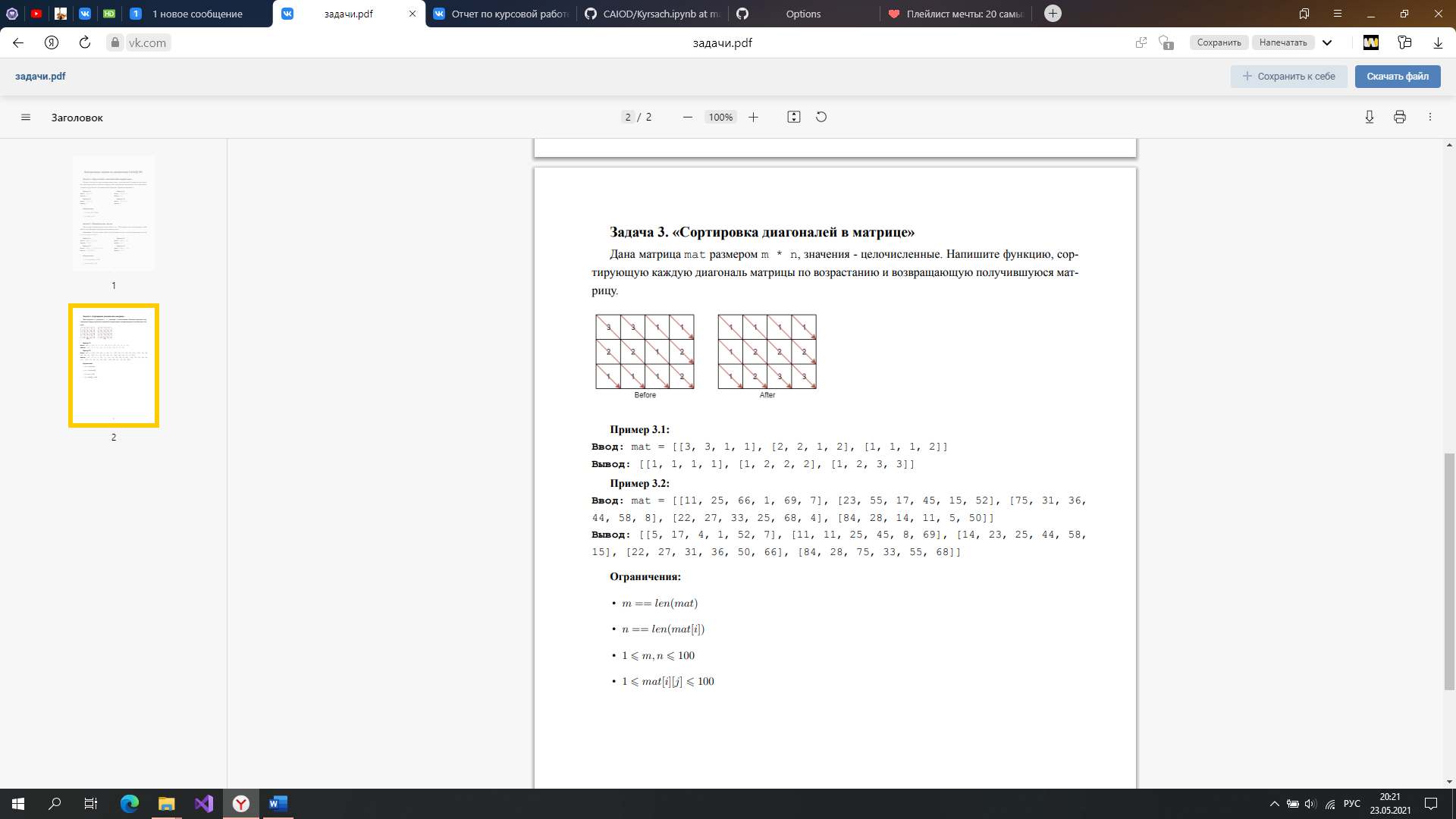
группы БВТ1902: Капленко Е. М.

Руководитель: Мкртчян Г. М.

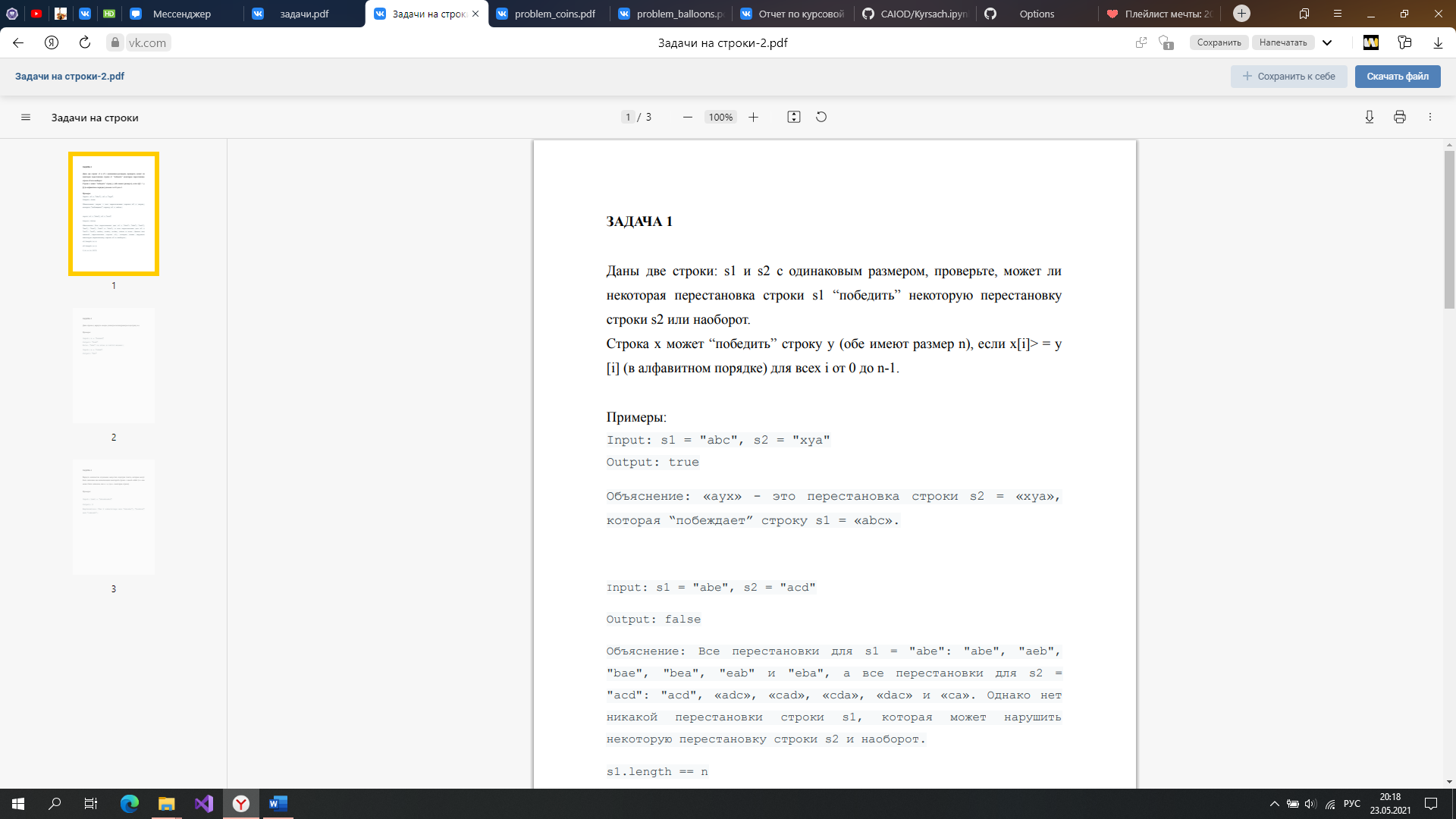
Москва 2020

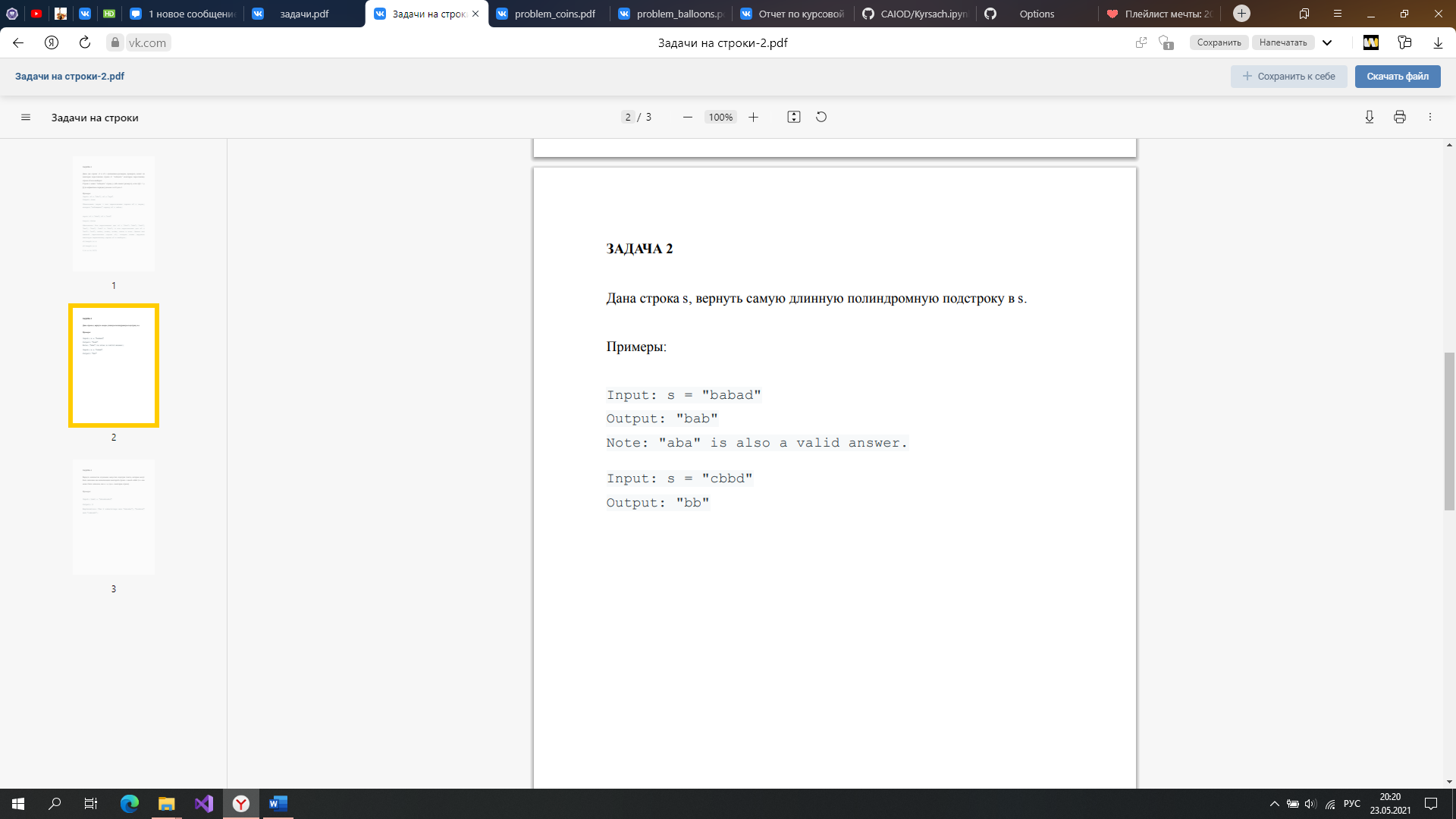


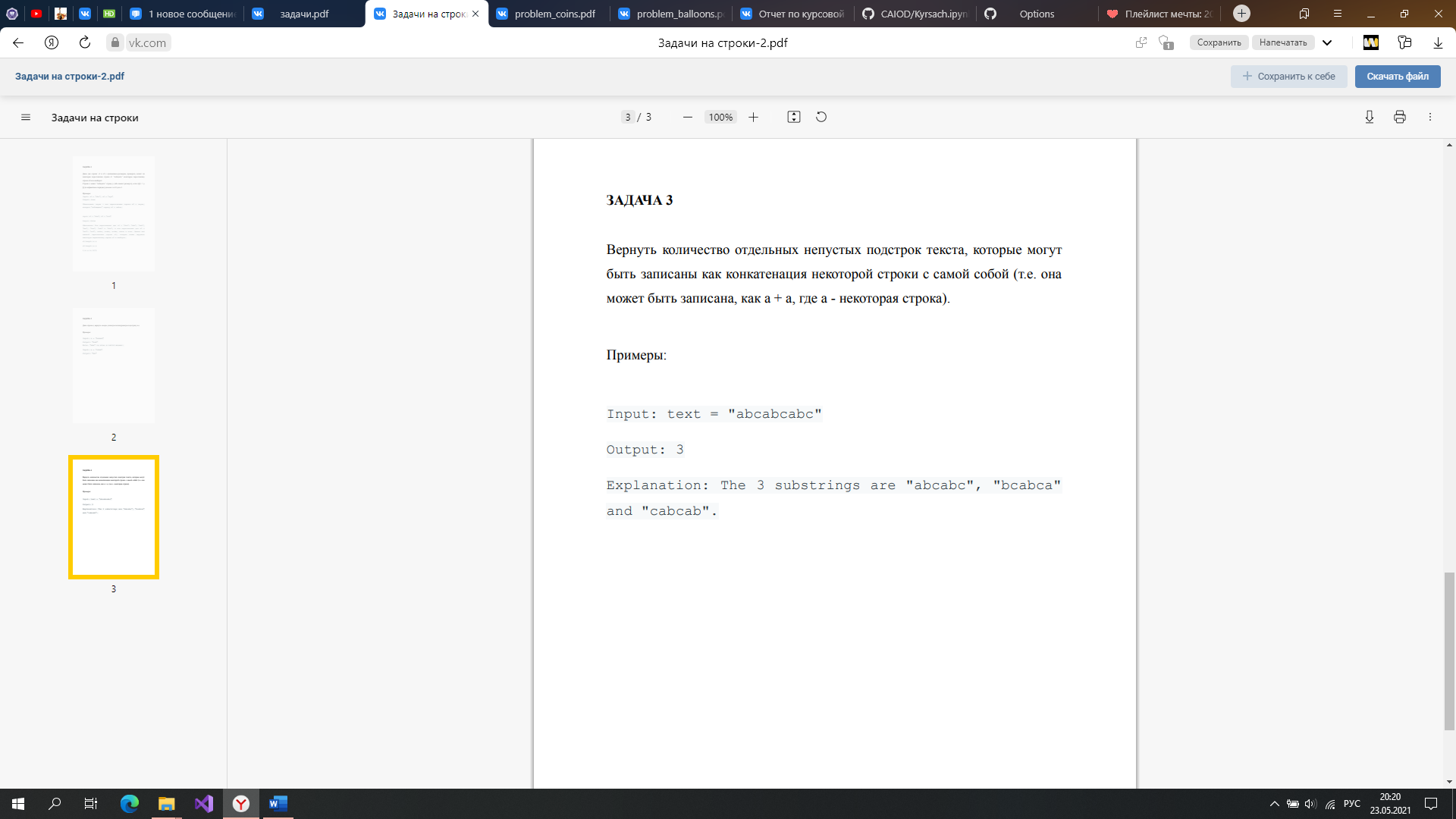


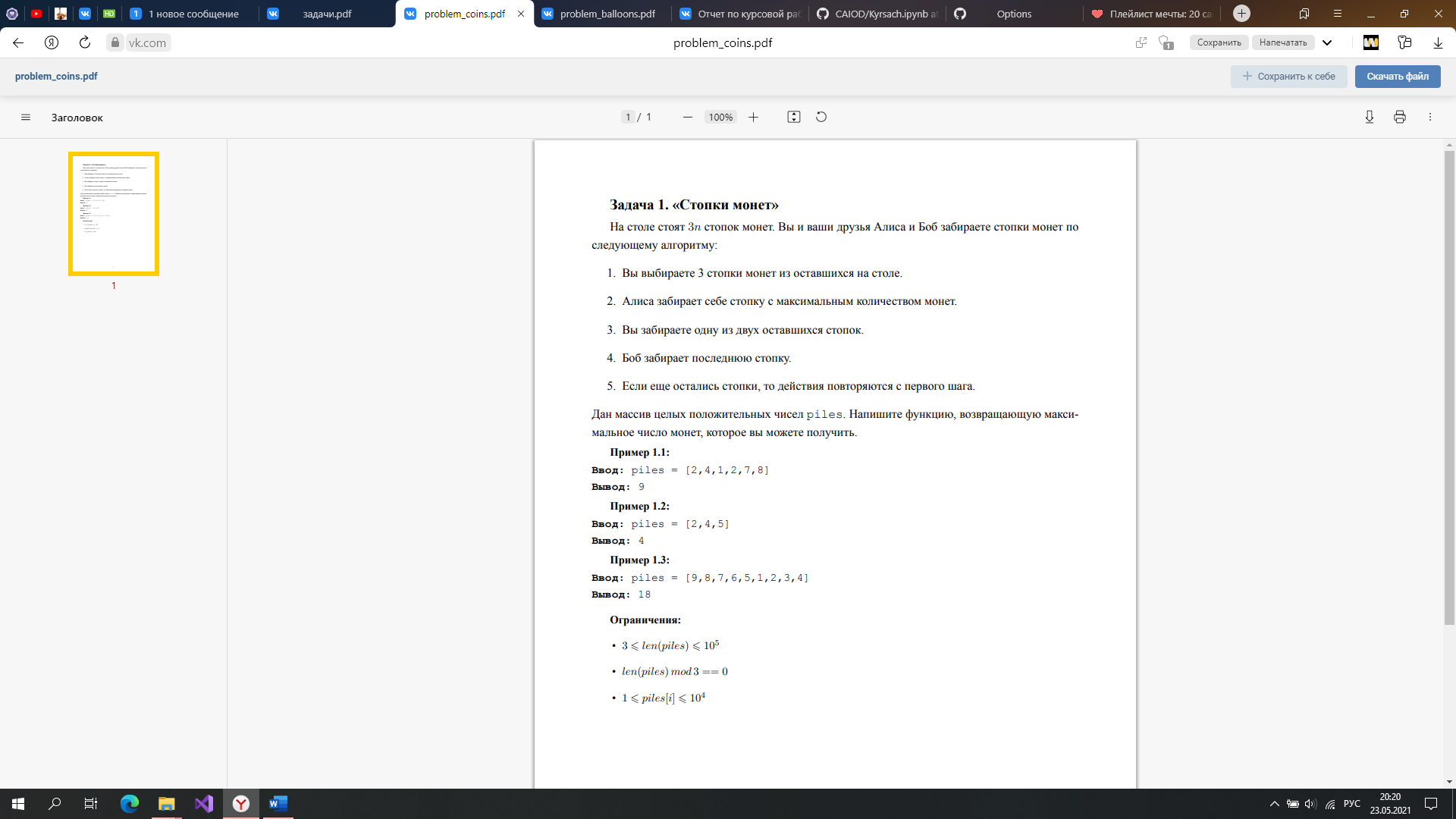


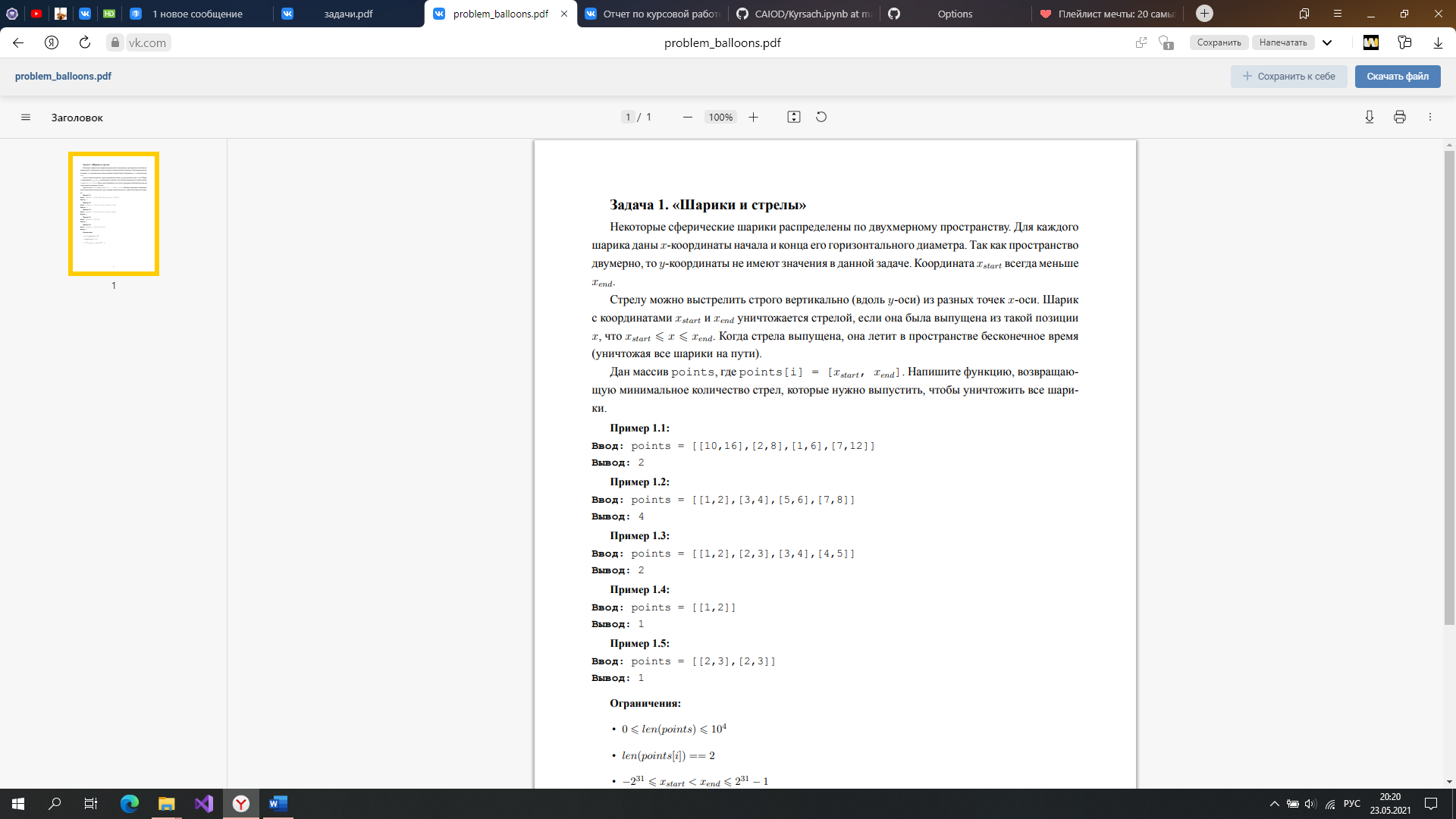
Задачи по строкам:











**Для решения задач написаны следующие функции:**

static int Task1(int[] arr) {  
 Arrays.sort(arr);  
 arr = forTask1(arr);  
 for(int i = 0; i < arr.length-2; i++)  
 {  
 if ((arr[i] < (arr[i+1] + arr[i+2])) && (arr[i+1] < (arr[i] + arr[i+2])) && (arr[i+2] < (arr[i+1] + arr[i])))  
 {  
 return (arr[i] + arr[i+1] + arr[i+2]);  
  
 }  
 }  
 return 0;  
}  
static int[] forTask1(int[] myArray){  
 int size = myArray.length;  
 for (int i = 0; i < size / 2; i++) {  
 int temp = myArray[i];  
 myArray[i] = myArray[size - 1 - i];  
 myArray[size - 1 - i] = temp;  
 }  
 return myArray;  
}  
  
static void Task2(int[] arr)  
{  
 int MatrixSize = arr.length;  
 int index = 0;  
  
 for (int i = 0; i < MatrixSize - 1; i++)  
 {  
 index = i;  
  
 for (int j = i + 1; j < MatrixSize; j++)  
 {  
 String str1 = Integer.toString(arr[j]) + Integer.toString(arr[index]);  
 String str2 = Integer.toString(arr[index]) + Integer.toString(arr[j]);  
  
 if (Integer.parseInt(str1) < Integer.parseInt(str2))  
 {  
 index = j;  
 }  
 }  
  
 if (index != i)  
 {  
 int temp = arr[i];  
 arr[i] = arr[index];  
 arr[index] = temp;  
 }  
 }  
}  
  
  
static void Task3(int[][] arr, int m,int n){  
 for(int i = 0; i < n-1; i++){  
 FuncForTask3(arr, 0,i,m,n);  
 }  
 for (int i = 1; i < m - 1; i++)  
 {  
 FuncForTask3(arr, i, 0, m, n);  
 }  
}  
  
static void FuncForTask3(int[][] arr, int m,int k,int lenX,int lenY) {  
 ArrayList<Integer> NewArr = new ArrayList<>();  
  
  
 int m1 = m;  
 int k1 = k;  
 while (ProvForTask3(m1,k1,lenX,lenY)) {  
 NewArr.add(arr[m1][k1]);  
 m1++;  
 k1++;  
 }  
  
 Collections.sort(NewArr);  
 int g = 0;  
 while (ProvForTask3(m, k, lenX, lenY)) {  
 arr[m][k]=NewArr.get(g);  
 m++;  
 k++;  
 g++;  
 }  
}  
  
static boolean ProvForTask3(int indexX,int indexY, int i,int j) {  
 if (indexX < i && indexY < j) return true;  
 else return false;  
 }  
  
static void Task4(ArrayList<ArrayList<Integer>> MyList) {  
 for (int i = 0; i < MyList.size()-1; i++) {  
 for (int j = i+1; j < MyList.size(); j++) {  
 if((MyList.get(i).get(0)>= MyList.get(j).get(0) && MyList.get(i).get(0) <= MyList.get(j).get(1))|| (MyList.get(j).get(0) >= MyList.get(i).get(0) && MyList.get(j).get(0) <= MyList.get(i).get(1))) {  
 MyList.remove(j);  
 }  
 }  
 }  
 System.out.println(MyList.size());  
}  
  
static void Task5(ArrayList<Integer> arr) {  
 Collections.sort(arr);  
 Collections.reverse(arr);  
 int sum = 0;  
 int n = 0;  
 int count = arr.size();  
 while (n!=count/3) {  
 arr.remove(0);  
 sum += arr.get(0);  
 arr.remove(0);  
 n++;  
 }  
 System.out.println(sum);  
}  
  
  
static void Task6(char[] ch1,char[]ch2) {  
  
 char[] ch3=ch1;  
 char[] ch4=ch2;  
 Arrays.sort(ch1);  
 Arrays.sort(ch2);  
  
 if(ch3==ch1) {  
 char temp = ch1[ch1.length-1];  
 ch1[ch1.length - 1] = ch1[ch1.length - 2];  
 ch1[ch1.length - 2] = temp;  
 }  
 if (ch4 == ch2) {  
 char temp = ch2[ch1.length - 1];  
 ch2[ch1.length - 1] = ch2[ch1.length - 2];  
 ch2[ch1.length - 2] = temp;  
 }  
  
 String st1 = "";  
 String st2 = "";  
  
 for(int i = 0; i < ch1.length; i++) {  
 st1 += ch1[i];  
 st2 += ch2[i];  
 }  
  
 if (st1.equals(st2) == true) {  
 System.out.println("Ни одна перестановка второй строки не победит первую строку");  
 } else {  
 System.out.println("Вторая строка побеждает");  
 }  
}  
  
  
static void Task7(String str8) {  
 int n = 0;  
 while (n != str8.length()) {  
 String s = str8.substring(0, str8.length() - n);  
 if (IsPalindrom(s)) {  
 System.out.println(s);  
 return;  
 }  
 n++;  
 }  
}  
static boolean IsPalindrom(String str) {  
 int len = str.length() / 2;  
  
 for(int i = 0; i < len; i++) {  
 if (str.charAt(i) != str.charAt(str.length() - 1 - i)) return false;  
 }  
 return true;  
}  
  
static void Task8(String str) {  
 int count = 0;  
 LinkedList<String> strings = new LinkedList<>();  
  
 for (int i = 0; i < str.length(); i++) {  
 int k = i;  
 int z = 1;  
 while (z < str.length()-i) {  
 String Str = str.substring(k, z);  
 String Mystr = Str + Str;  
 if (str.contains(Mystr) && !strings.contains(Mystr)) {  
 count++;  
 strings.add(Mystr);  
 }  
 z = z + 1;  
 }  
 }  
 System.out.println(count);  
}

**Вывод:** Задачи успешно выполнены, при решении данных задач познакомилась с использованием листов с вложенными листами.