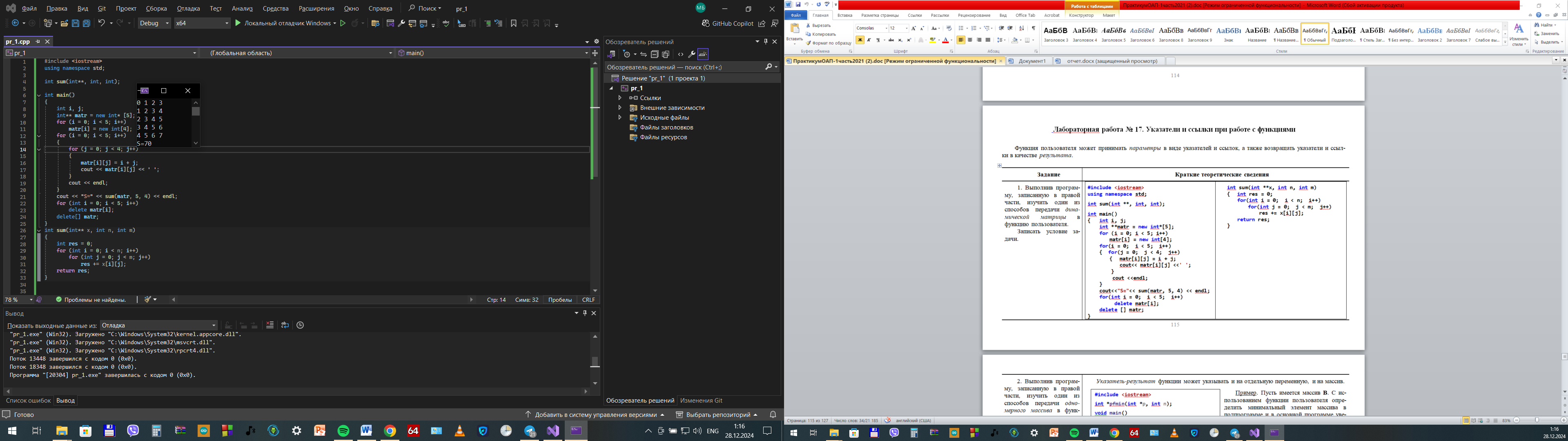
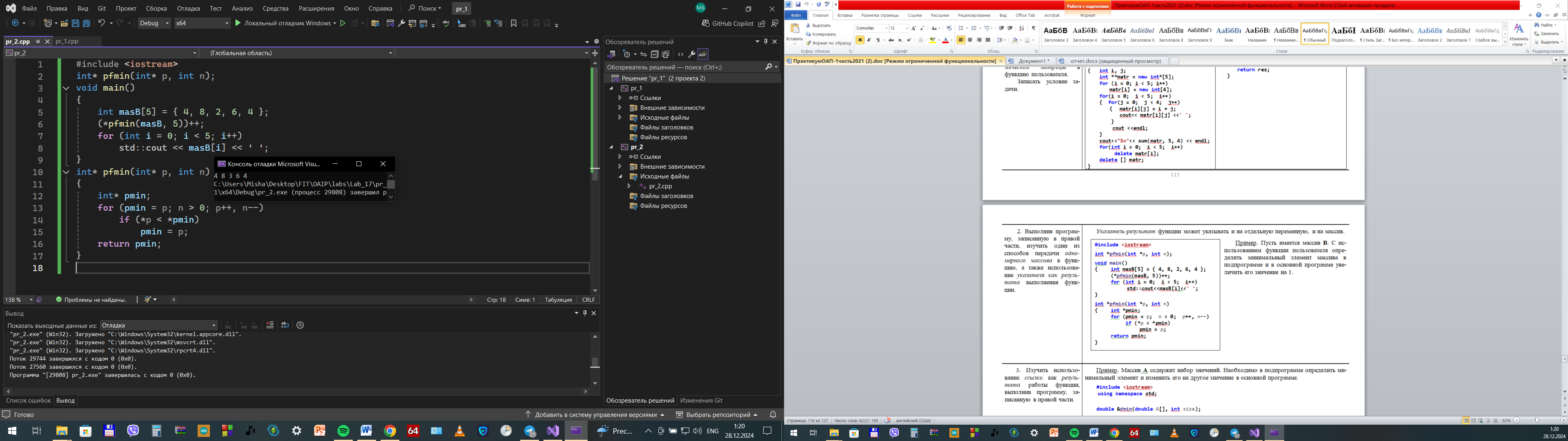
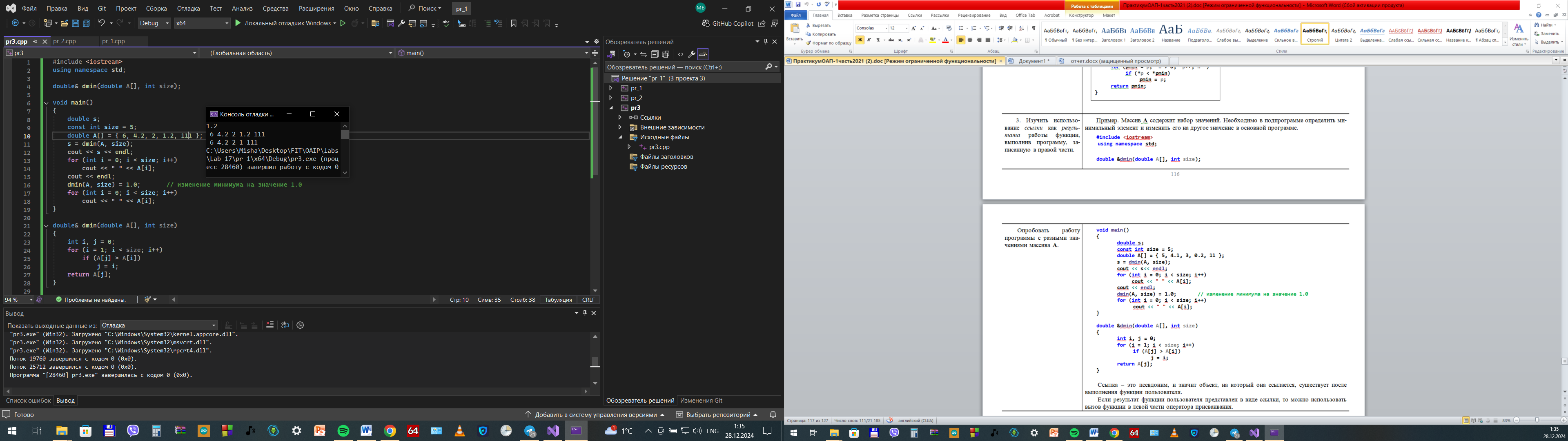
Pr\_1 Условие: Посчитать сумму всех элементов матрицы



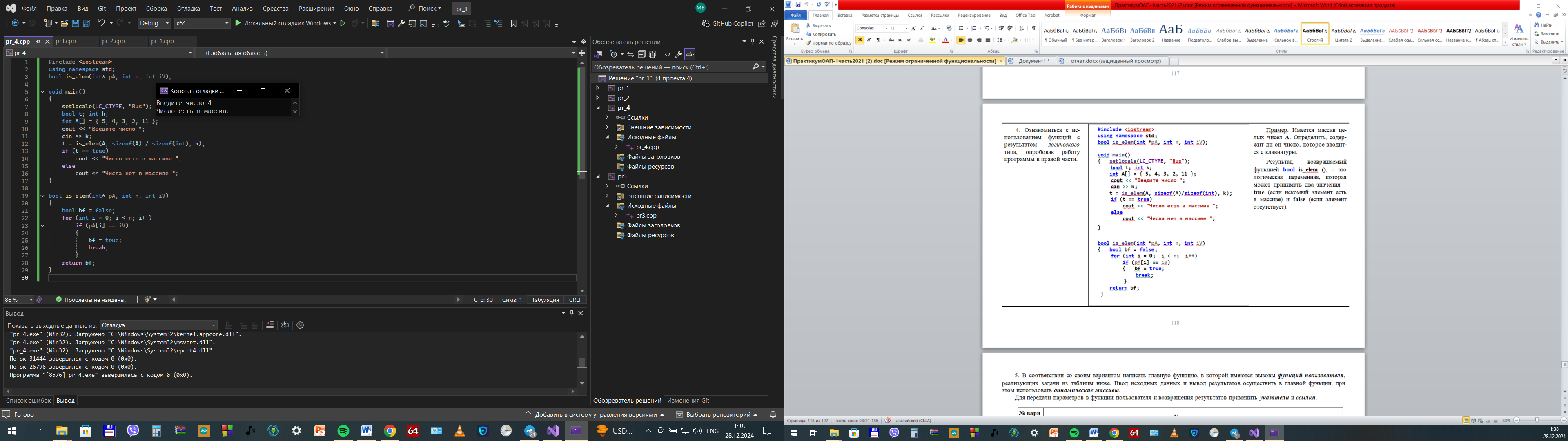
Pr\_2



Pr\_3



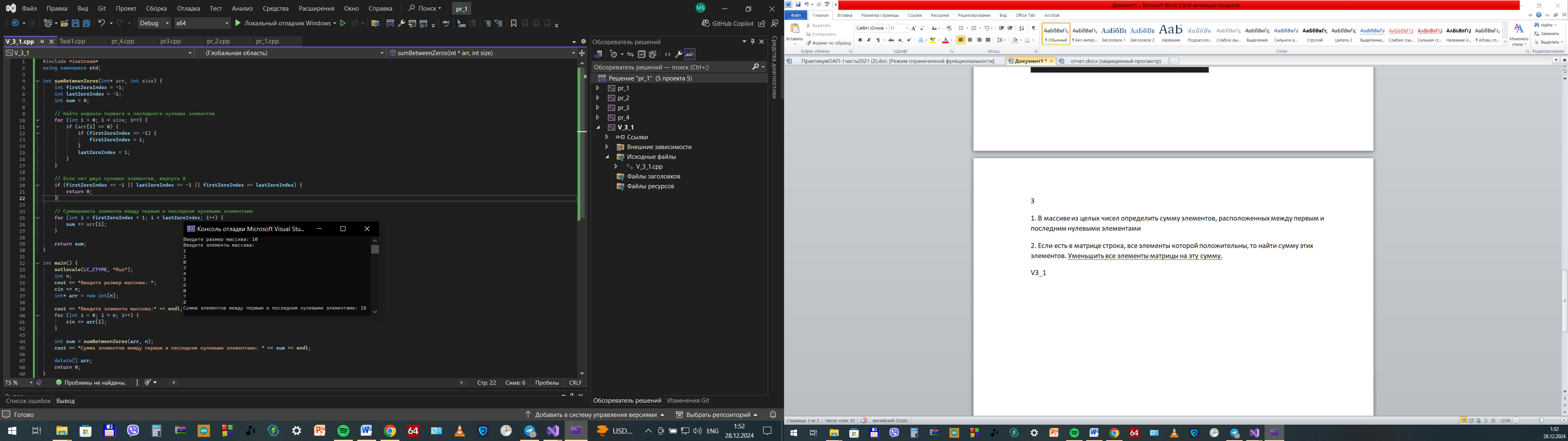
Pr\_4



V3\_1

1. В массиве из целых чисел определить сумму элементов, расположенных между первым и последним нулевыми элементами

2. Если есть в матрице строка, все элементы которой положительны, то найти сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.



V3\_2

2. Если есть в матрице строка, все элементы которой положительны, то найти сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.



#include <iostream>

using namespace std;

void processMatrix(int\*\* matrix, int rows, int cols) {

int rowSum = 0;

bool positiveRowFound = false;

for (int i = 0; i < rows; i++) {

bool allPositive = true;

for (int j = 0; j < cols; j++) {

if (matrix[i][j] <= 0) {

allPositive = false;

break;

}

}

if (allPositive) {

positiveRowFound = true;

for (int j = 0; j < cols; j++) {

rowSum += matrix[i][j];

}

cout << "Сумма элементов положительной строки (" << i + 1 << "): " << rowSum << endl;

break;

}

}

if (positiveRowFound) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

matrix[i][j] -= rowSum;

}

}

}

}

int main() {

setlocale(LC\_CTYPE, "Rus");

int rows, cols;

cout << "Введите количество строк и столбцов матрицы: ";

cin >> rows >> cols;

int\*\* matrix = new int\* [rows];

for (int i = 0; i < rows; i++) {

matrix[i] = new int[cols];

}

cout << "Введите элементы матрицы: " << endl;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cin >> matrix[i][j];

}

}

cout << "Исходная матрица: " << endl;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

processMatrix(matrix, rows, cols);

cout << "Матрица после обработки: " << endl;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < rows; i++) {

delete[] matrix[i];

}

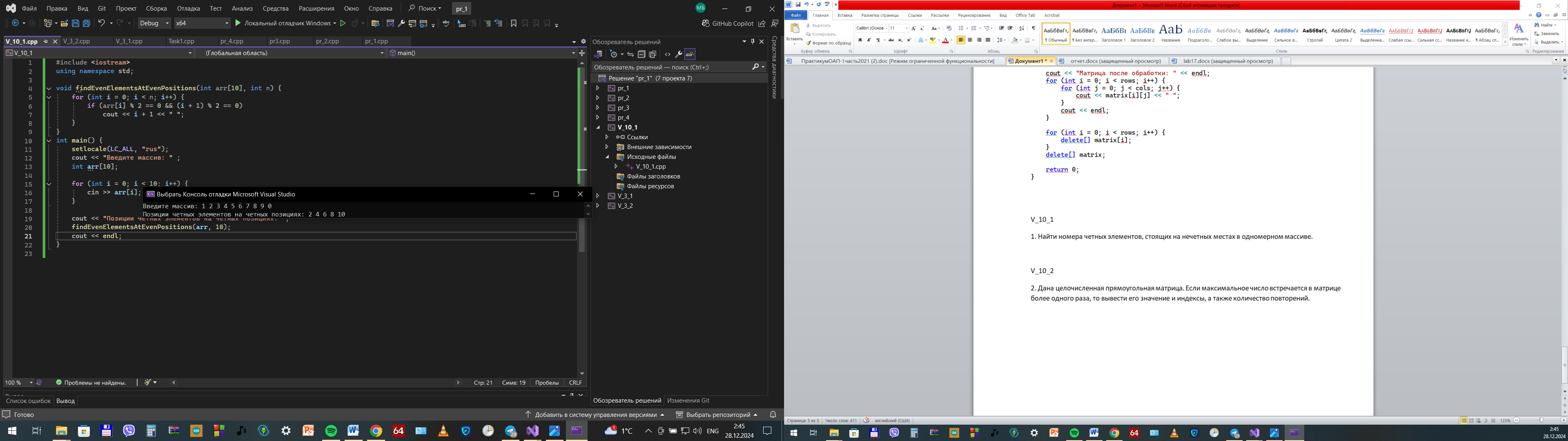
delete[] matrix;

return 0;

}

V\_10­\_1

1. Найти номера четных элементов, стоящих на нечетных местах в одномерном массиве.



V\_10­\_2

2. Дана целочисленная прямоугольная матрица. Если максимальное число встречается в матрице более одного раза, то вывести его значение и индексы, а также количество повторений.

#include <iostream>

using namespace std;

float ret[4];

float findmax(float\*\* arr, int n, int m)

{

int max = arr[0][0], maxi = 0, maxj = 0, count = 1;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

if (arr[i][j] > max)

{

max = arr[i][j];

maxi = i;

maxj = j;

count = 1;

}

else if (arr[i][j] == max)

{

count++;

}

}

}

ret[0] = max;

ret[1] = maxi;

ret[2] = maxj;

ret[3] = count;

return 0;

}

int main()

{

setlocale(LC\_ALL, "rus");

int n, m;

cout << "Введите размер матрицы: ";

cin >> n >> m;

float\*\* arr;

arr = new float\* [n];

for (int i = 0; i < n; i++)

arr[i] = new float[m];

cout << "Введите матрицу: \n";

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cin >> arr[i][j];

}

}

findmax(arr, n, m);

float max = ret[0], maxi = ret[1], maxj = ret[2], count = ret[3];

cout << "Максимальный элемент: " << max << "[" << maxi + 1 << "]" << "[" << maxj + 1 << "]" << "\nОн встречается " << count << " раз"; // Вывод

for (int i = 0; i < n; i++)

delete[] arr[i];

delete[] arr;

}

