**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление информатика и вычислительная техника

Отчет

по лабораторной работе №4

по дисциплине

**«**Программирование**»**

**Обработка многомерных динамических массивов**

Вариант - 13

Выполнил:

Студент группы 8В31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Моцыборова А.А.

Проверил:

Программист \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Воеводин К.Д.

Цель работы

Познакомиться с работой с динамической памятью.

Задание

1.Составить алгоритм решения задачи в соответствии с заданным вариантом.

2.На основе созданного алгоритма разработать приложение.

3.Произвести тестирование разработанного приложения.

Должен быть обеспечен ввод размеров массива (и значений ячеек массива) пользователем. Должна быть обеспечена корректная работа с динамической памятью. Чтение данных организовать с клавиатуры (по согласованию с преподавателем – из файла), вывод результата – в консоль. Предусмотреть форматированный вывод данных в удобном для пользователя виде.

В прямоугольной матрице определить второй по величине элемент и его местоположение в массиве.

Программа должна быть разбита на три последовательных логических блока: чтение и запись данных в память, обработка данных, вывод результатов.

Ход работы

На рисунках 1 – 4 демонстрируется работа программы

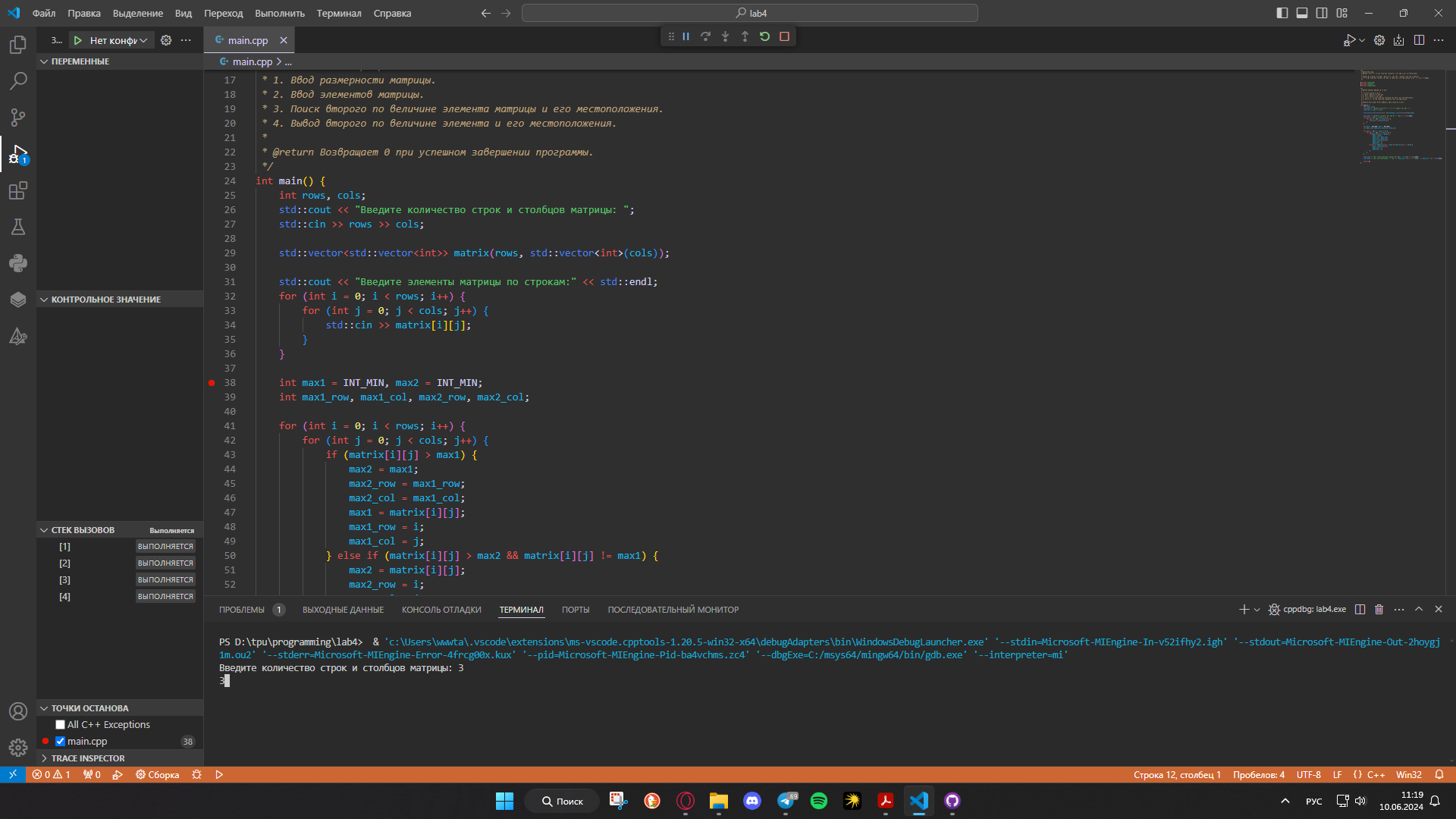


Рисунок 1 – Ввод парметров матрицы

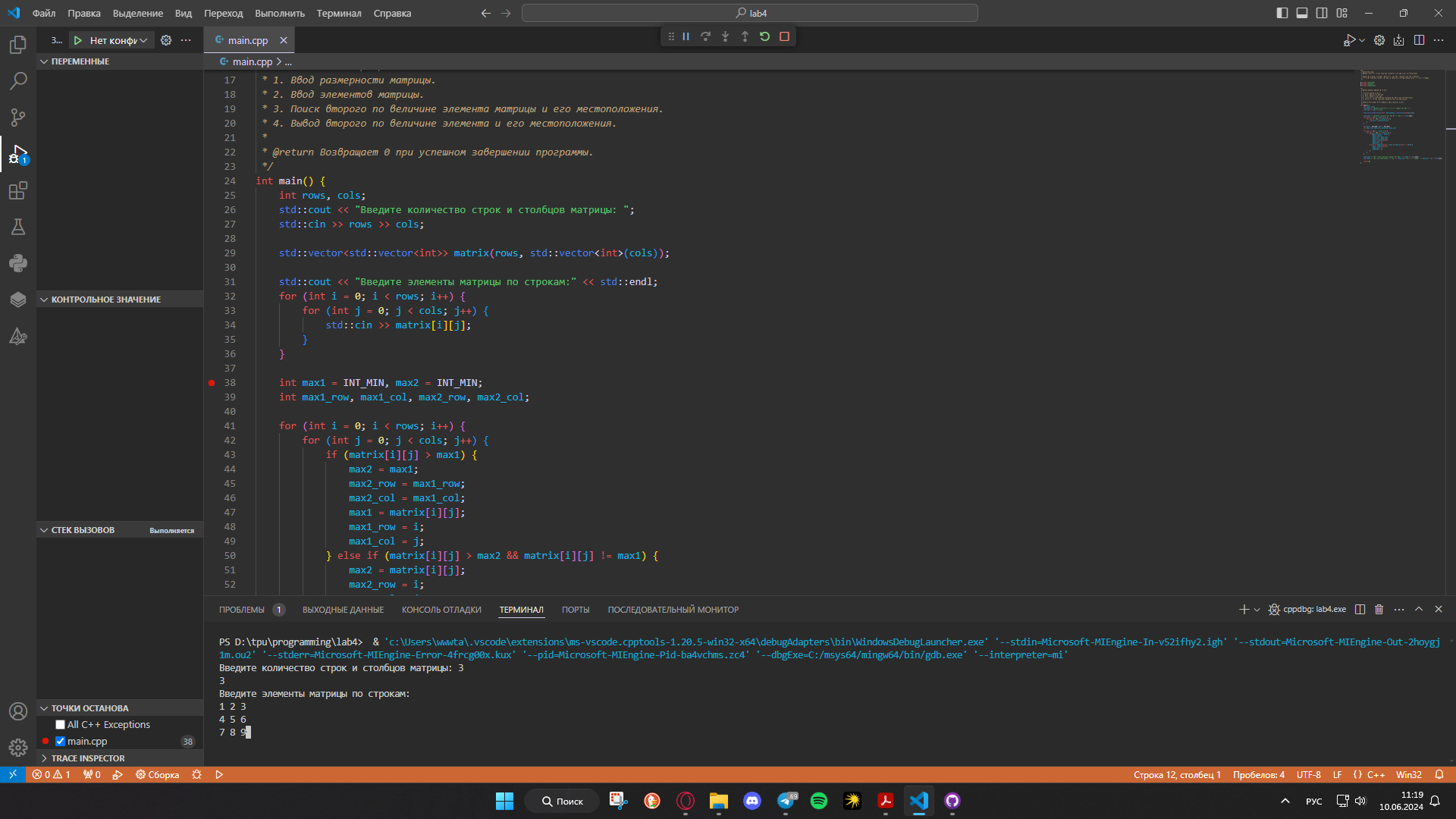
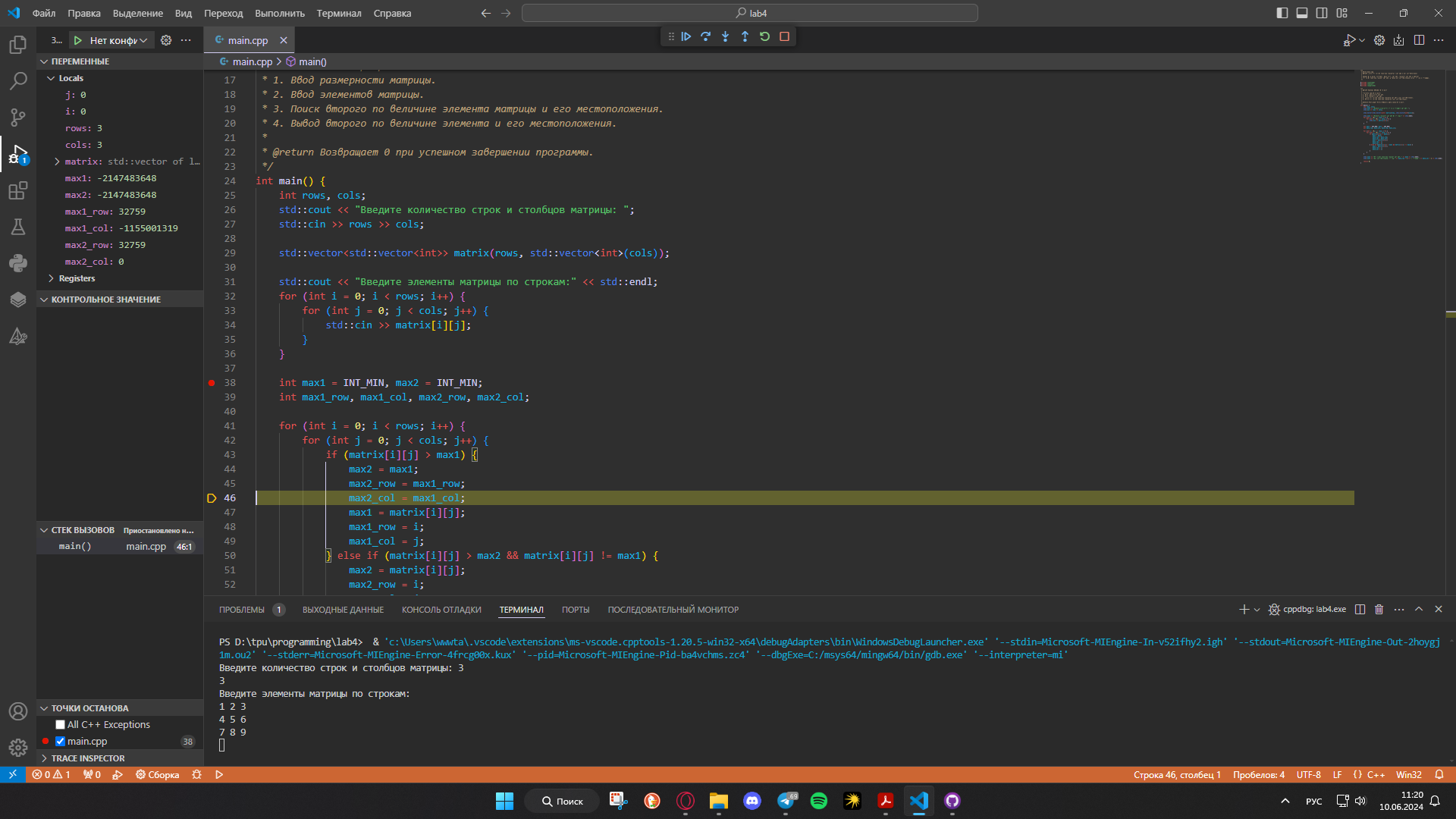


Рисунок 2 –Ввод матрицы

Рисунок 3 – Поиск искомого элемента в цикле for

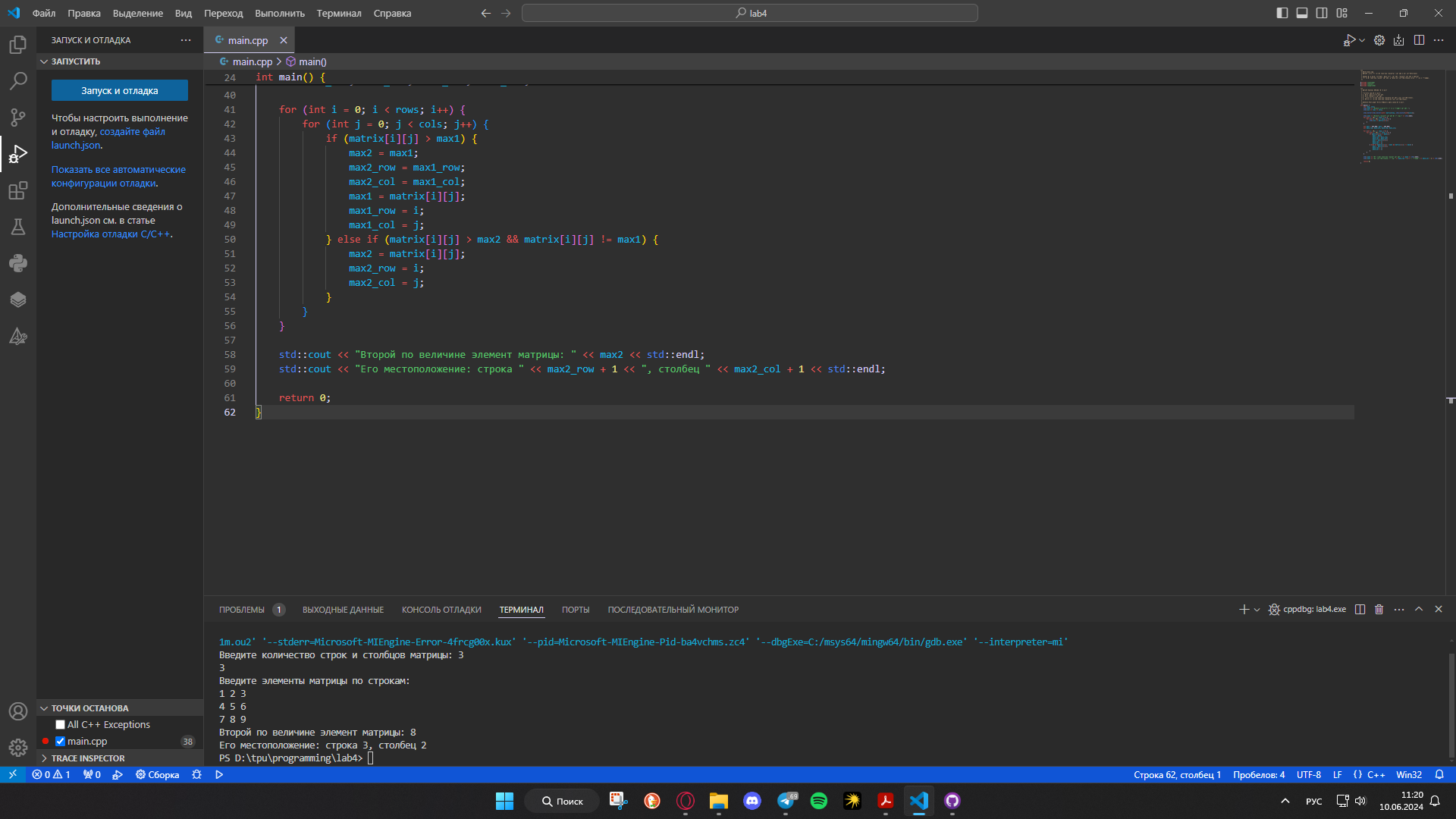


Рисунок 4 – Вывод искомого элемента

Вывод

Познакомились с работой с динамической памятью.Приложение 1 – Код программы

/\*\*

\* @file main.cpp

\* @brief Поиск второго по величине элемента в матрице и его местоположения

\*

\* Данная программа считывает размерность матрицы, элементы матрицы и находит

\* второй по величине элемент матрицы, а также его местоположение (номер строки и столбца).

\*/

#include <iostream>

#include <vector>

#include <algorithm>

/\*\*

\* @brief Главная функция программы

\*

\* Основные шаги программы:

\* 1. Ввод размерности матрицы.

\* 2. Ввод элементов матрицы.

\* 3. Поиск второго по величине элемента матрицы и его местоположения.

\* 4. Вывод второго по величине элемента и его местоположения.

\*

\* @return Возвращает 0 при успешном завершении программы.

\*/

int main() {

int rows, cols;

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

std::cin >> rows >> cols;

std::vector<std::vector<int>> matrix(rows, std::vector<int>(cols));

std::cout << "Введите элементы матрицы по строкам:" << std::endl;

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

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

std::cin >> matrix[i][j];

}

}

int max1 = INT\_MIN, max2 = INT\_MIN;

int max1\_row, max1\_col, max2\_row, max2\_col;

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

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

if (matrix[i][j] > max1) {

max2 = max1;

max2\_row = max1\_row;

max2\_col = max1\_col;

max1 = matrix[i][j];

max1\_row = i;

max1\_col = j;

}

else if (matrix[i][j] > max2 && matrix[i][j] != max1) {

max2 = matrix[i][j];

max2\_row = i;

max2\_col = j;

}

}

}

std::cout << "Второй по величине элемент матрицы: " << max2 << std::endl;

std::cout << "Его местоположение: строка " << max2\_row + 1 << ", столбец " << max2\_col + 1 << std::endl;

return 0;

}Ссылка на GitHub: <https://github.com/G1djet/Prog-lab4.git>