Отчет к лабораторной работе по предмету

«Конструирование программного обеспечения»

Подготовила

Гурина Кристина 1-4

Стек вызовов

Цель работы: изучить стек вызовов.

Задание 1.

Дополните функциональность приложения из первой лабораторной работы, добавив две функции:

1. связывающую порядковый номер месяца с его названием;

2. вычисляющую, на какую дату года приходится день программиста.

Дату вводить в формате ДДММГГГГ. Результаты выполнения выводить в консоль (месяц в текстовом виде).

1. Алгоритм функция связывающей порядковый номер месяца с его названием представлен на рисунку 1

функция Month



Рисунок 1

В функцию Month передаётся 2 параметра: день и месяц. Для реализации создана строка names на 12 элементов. Связывание порядкового номера с названием месяца – names[month - 1]. Минус 1 необходима так как нумерация в массиве начинается с 0.

Код функции представлен ниже:

int Month(int day, int month) {

string names[12] = { "Января", "Февраля", "Марта", "Апреля", "Мая", "Июня", "Июля", "Августа", "Сентября", "Октября", "Ноября", "Декабря" };

cout << day << " " << names[month - 1];

return 0;

}

Результат работы функции Month представлен на рисунке 2.

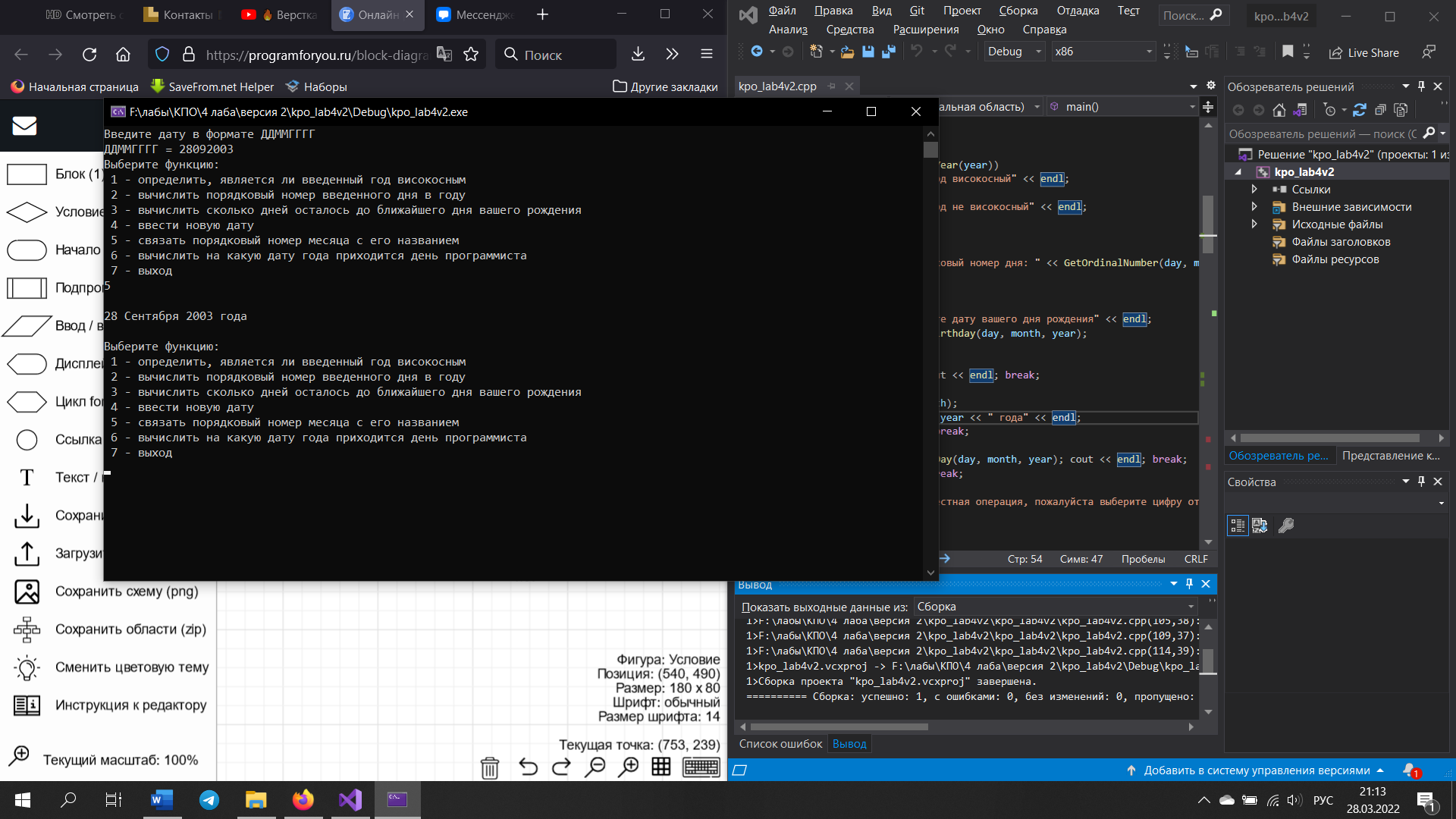


Рисунок 2

1. Алгоритм функция, вычисляющей на какую дату года приходится день программиста представлен на рисунку 3.

Функция ProgrammerDay

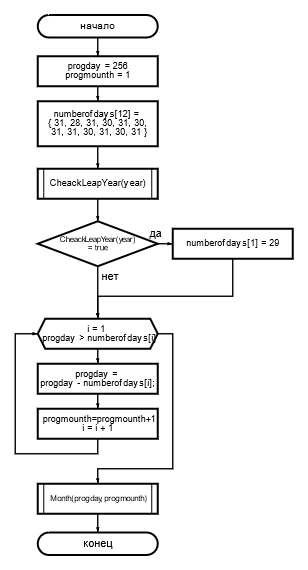


Рисунок 3

Инициализируем переменную progday значением 256 (порядковый номер дня программиста в году), а переменную progmounth – значением 1.

Создаём целочисленный массив numberofdays на 12 элементов. Элементами массива является количество дней в каждом месяце. Далее вызываем функцию CheackLeapYear(year) для проверки является ли год високосным. Если функция возвращает значение true, то numberofdays[1] = 29.

Создаём цикл for со счётчиком i = 0, условием выхода из цикла progday > numberofdays[i] и i++. Во время каждой итерации цикла от progday отнимается numberofdays[i]. Таким образом progday уменьшается на количество дней соответствующее i-тому элементу в массиве numberofdays. Переменная progmounth увеличивается на 1 с каждой итерацией и является счётчиком месяцев. По завершению цикла переменная progmounth показывает в каком месяце будет день программиста, а переменная progday – день на который он приходится.

Код функции представлен ниже:

int ProgrammerDay(int day, int month, int year) {

int progday = 256;

int progmounth = 1;

int numberofdays[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

if (CheackLeapYear(year))

numberofdays[1] = 29;

for (int i = 0; progday > numberofdays[i]; i++) {

progday = progday - numberofdays[i];

progmounth++;

}

cout << "В " << year << " году день программиста приходится на ";

Month(progday, progmounth);

cout << endl;

return 0;

}

Результат работы при вводе не високосного года представлена на рисунке 4, а при високосном – на рисунке 5.

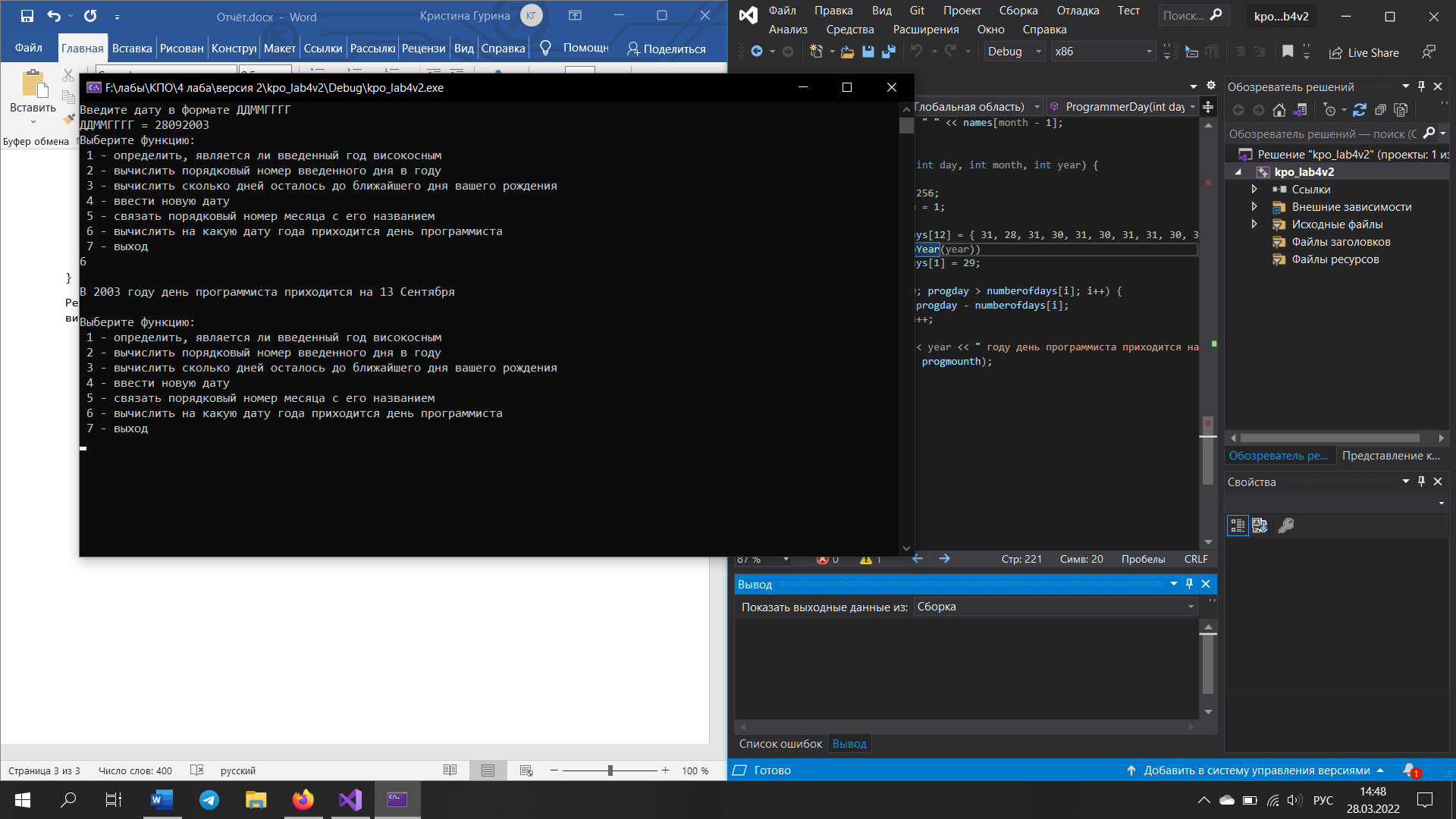


Рисунок 4

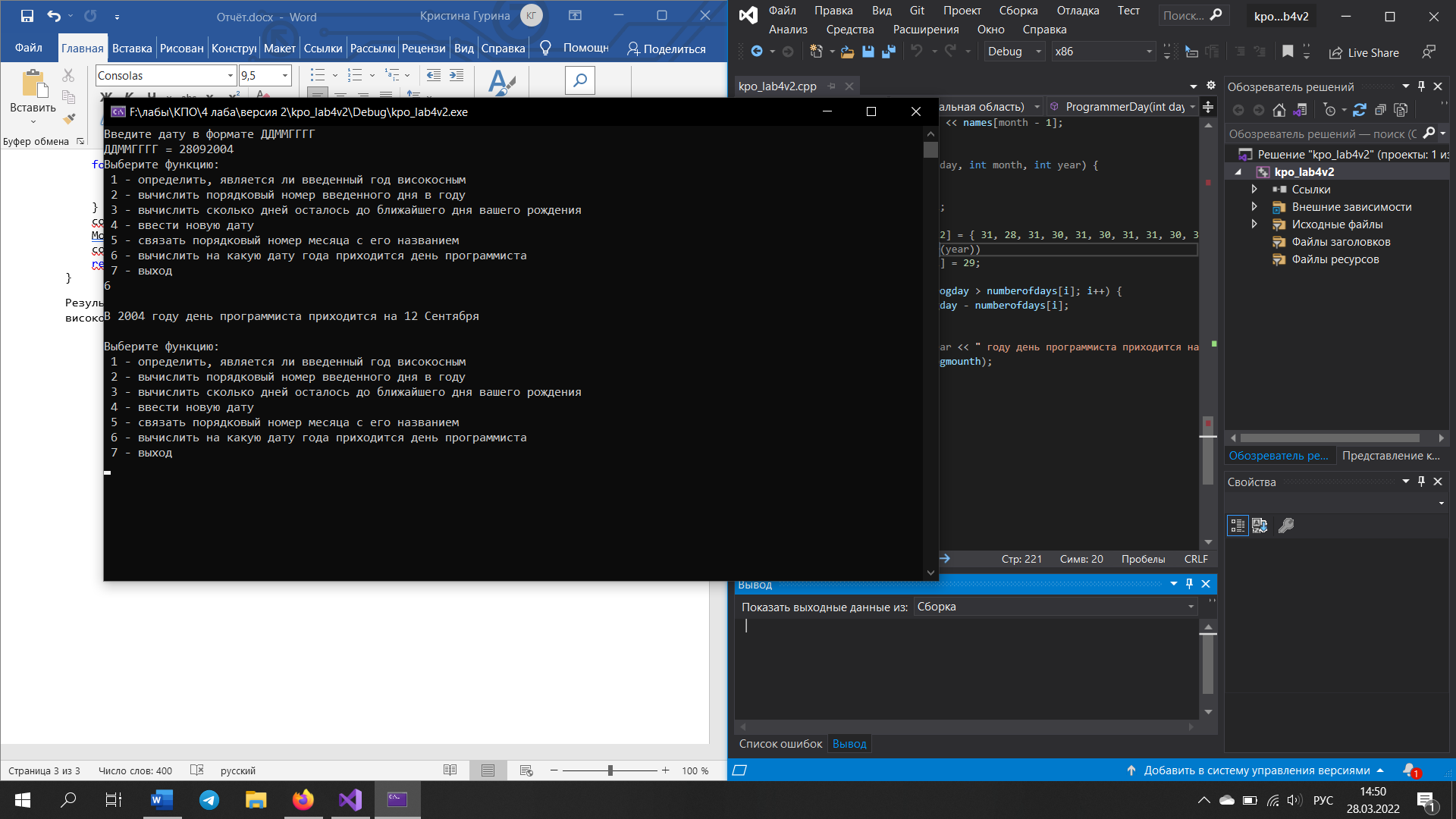
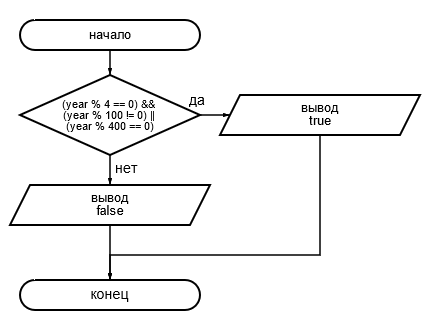
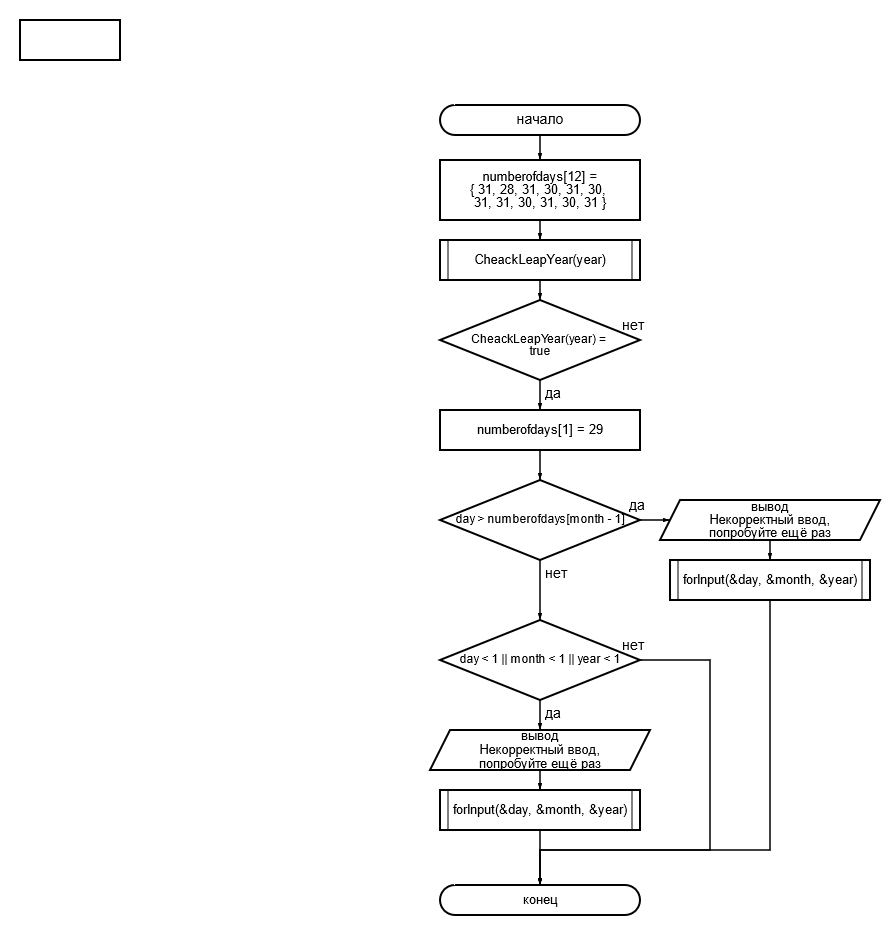


Рисунок 5

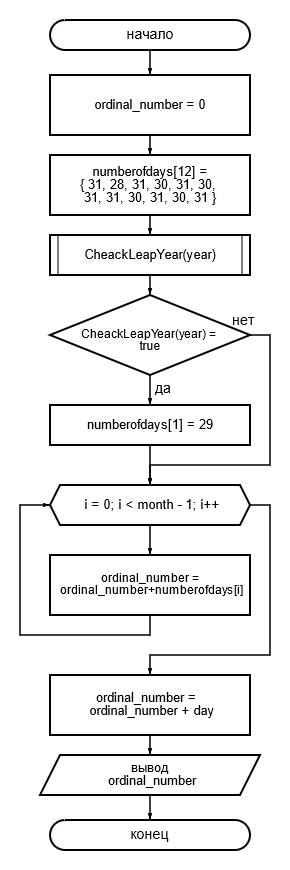
Функция CheackLeapYear



Функция CheackValidnost



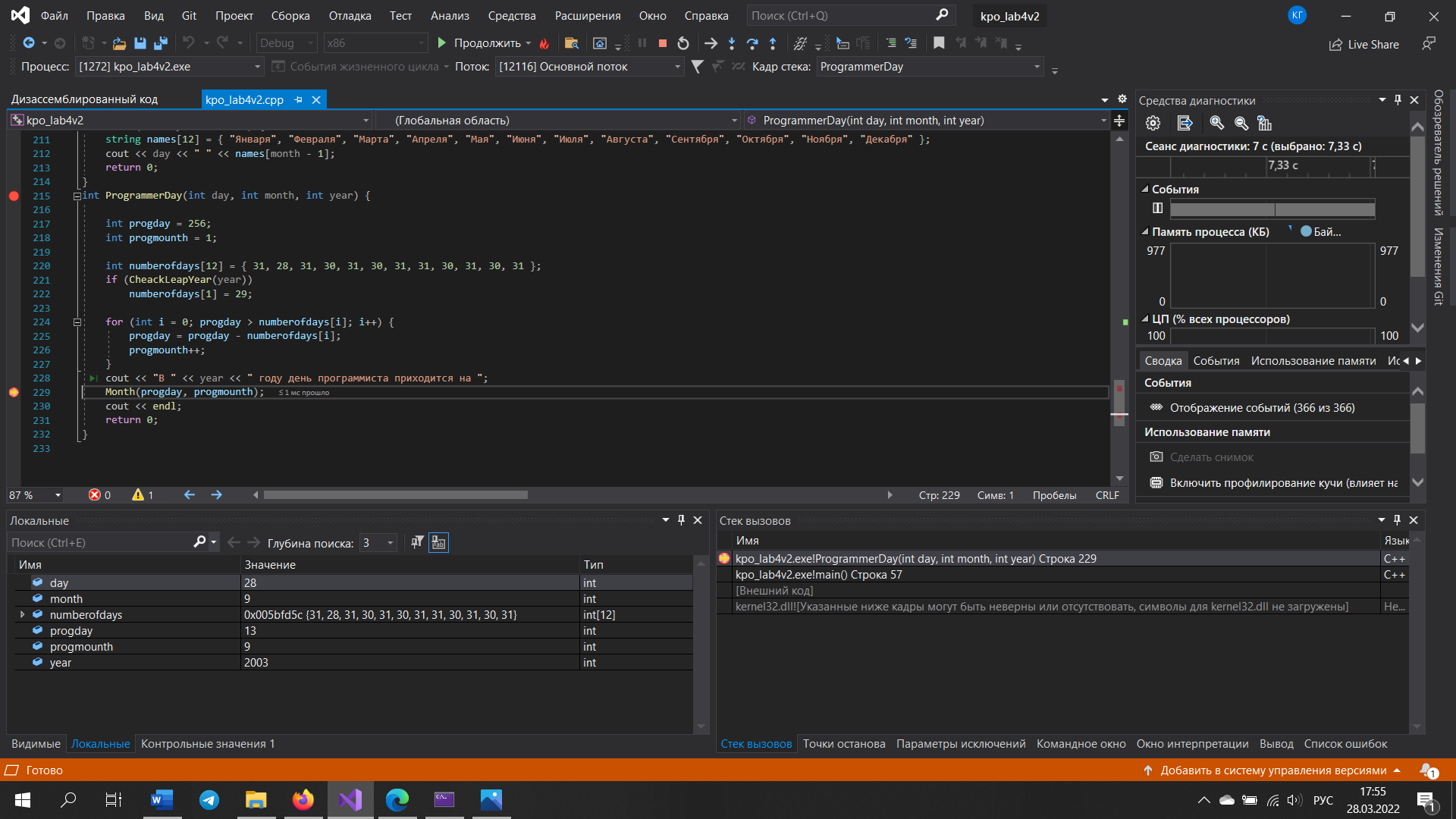
GetOrdinalNumber



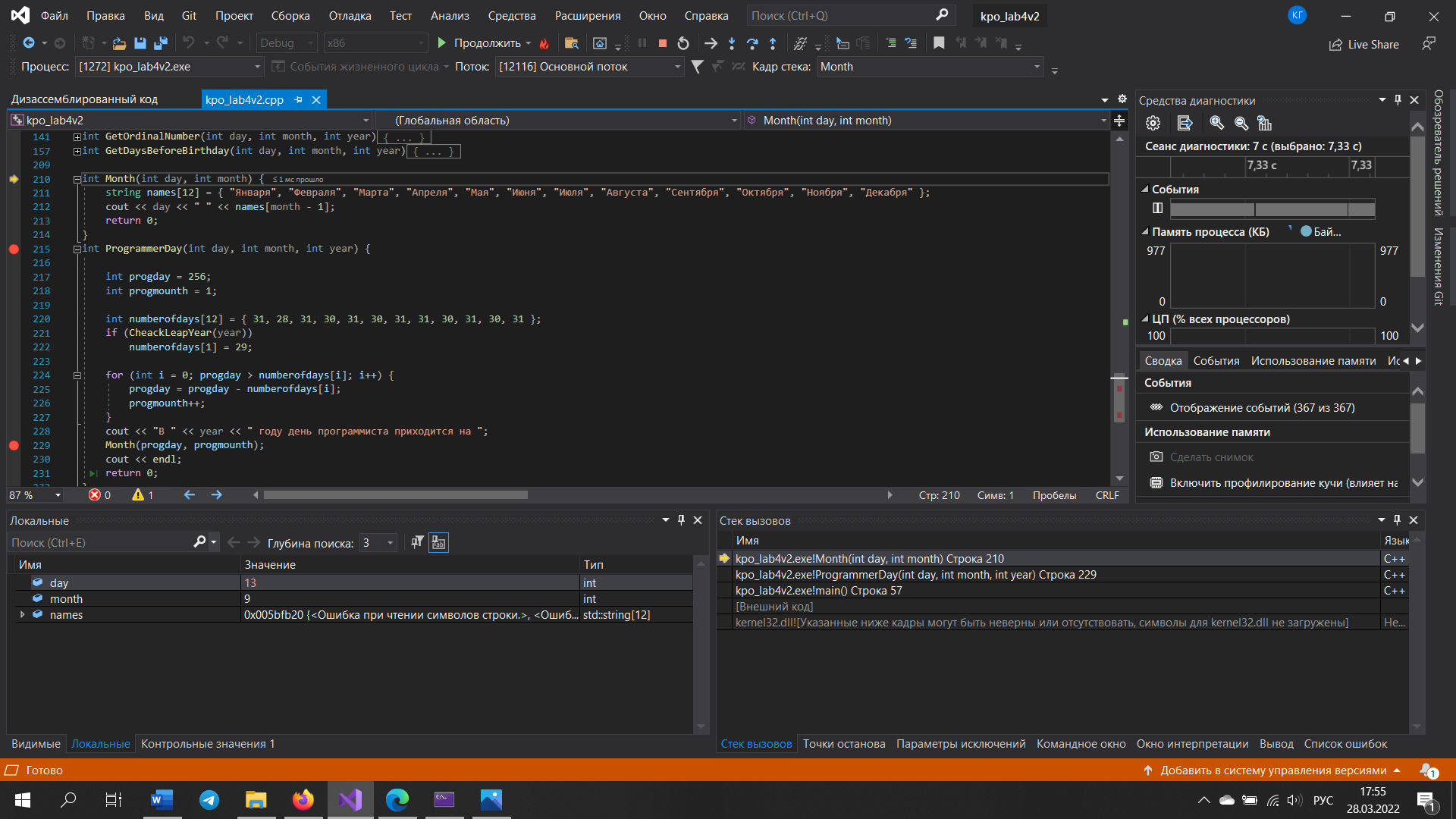
Задание 2.

Выполните приложение в пошаговом режиме отладки: первый раз без захода в функцию, второй раз с заходом в функцию.

Первый раз приложение было выполнено в пошаговом режиме отладки без захода в функцию(F10), а второй раз с заходом в функцию (F11)

1. до захода в функцию

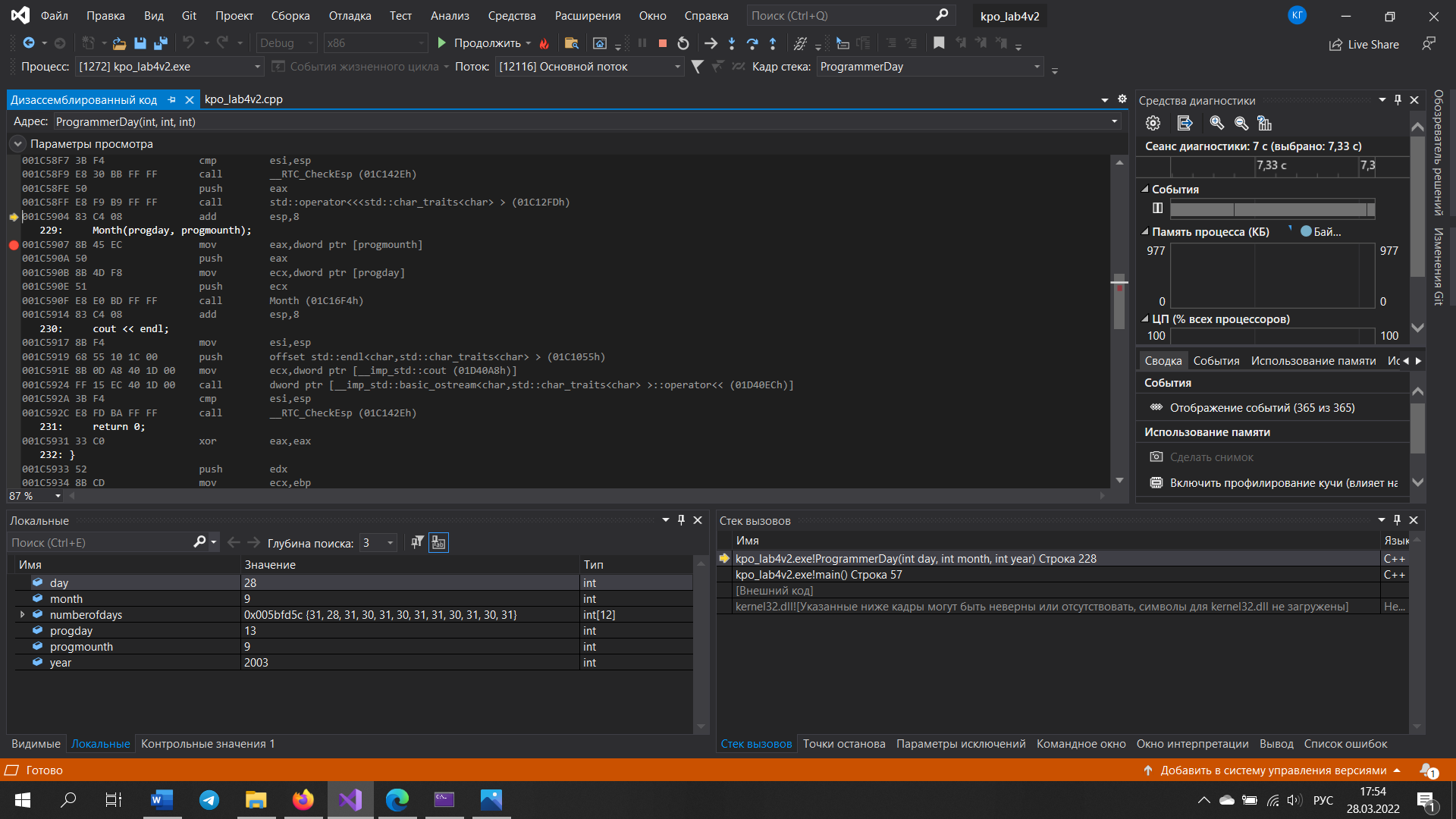
2. после захода в функцию

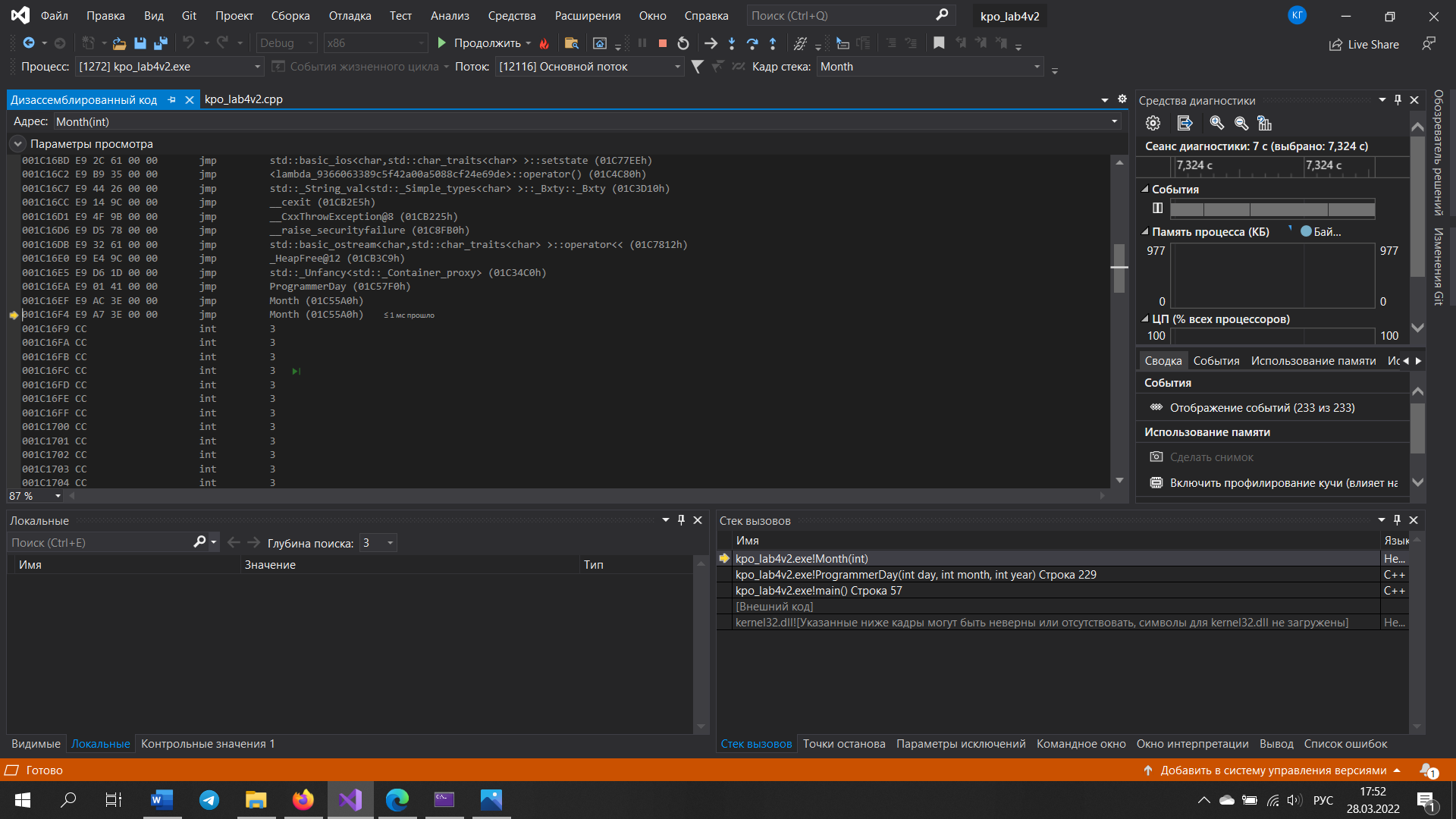


В результате захода в функцию мы видим, что стрелка, указывающая на строку, которая будет выполняться следующей, переместилась на функцию Month.

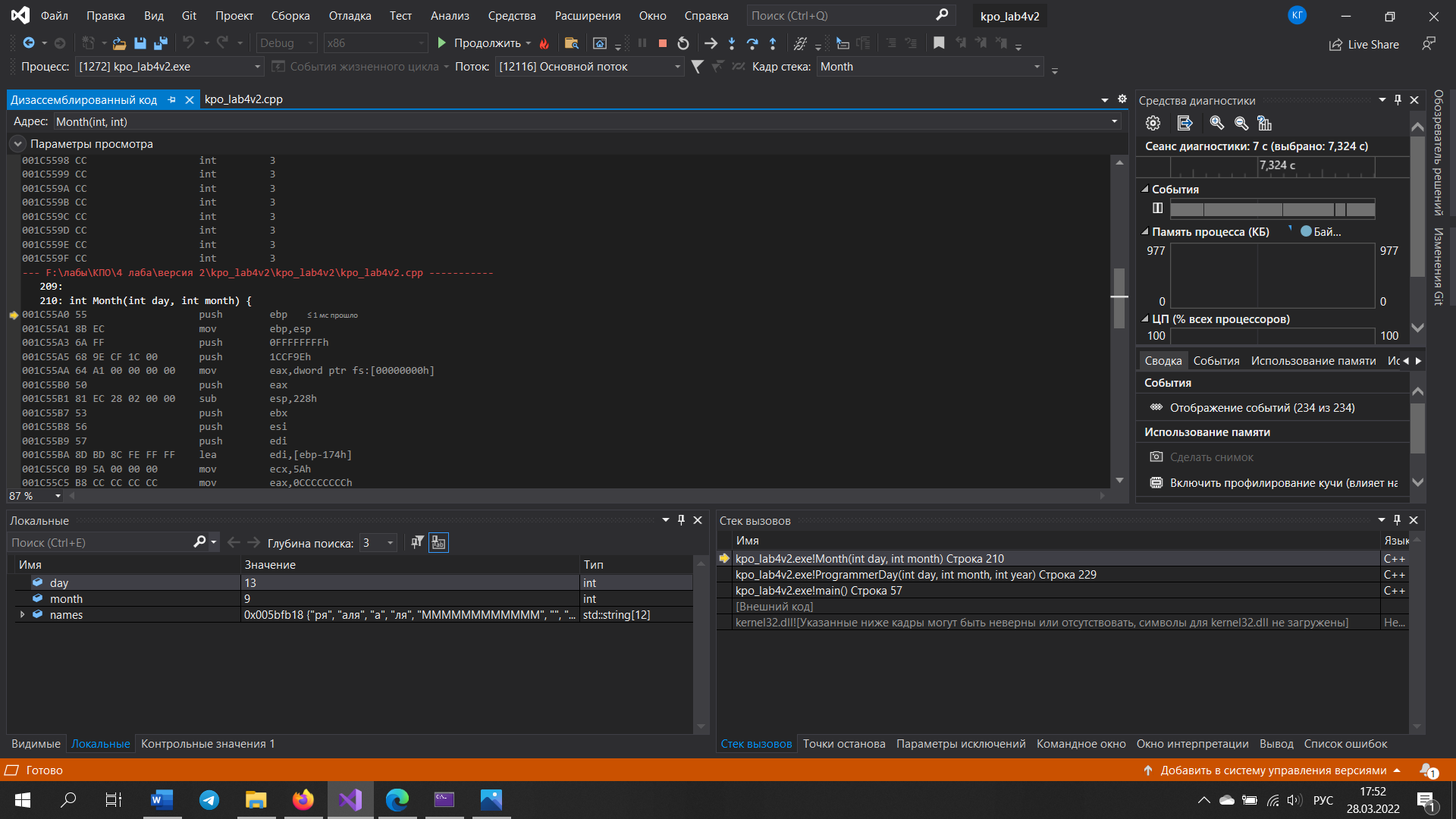
Рассмотрим пошаговое выполнение программы с заходом в функцию в дизассемблированном коде.

1. До захода в функцию:

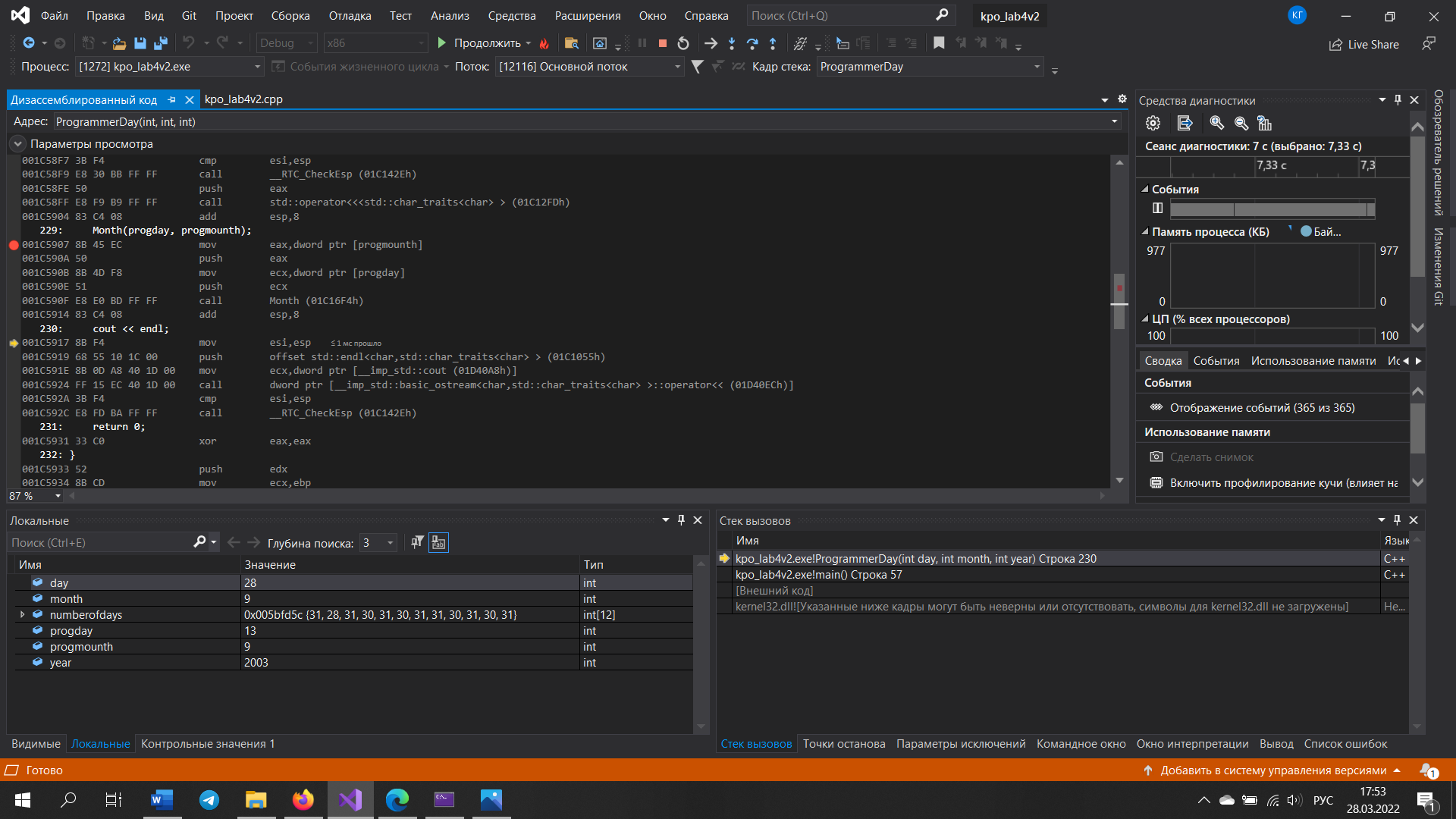


2. После захода в функцию.

3. После захода в функцию.

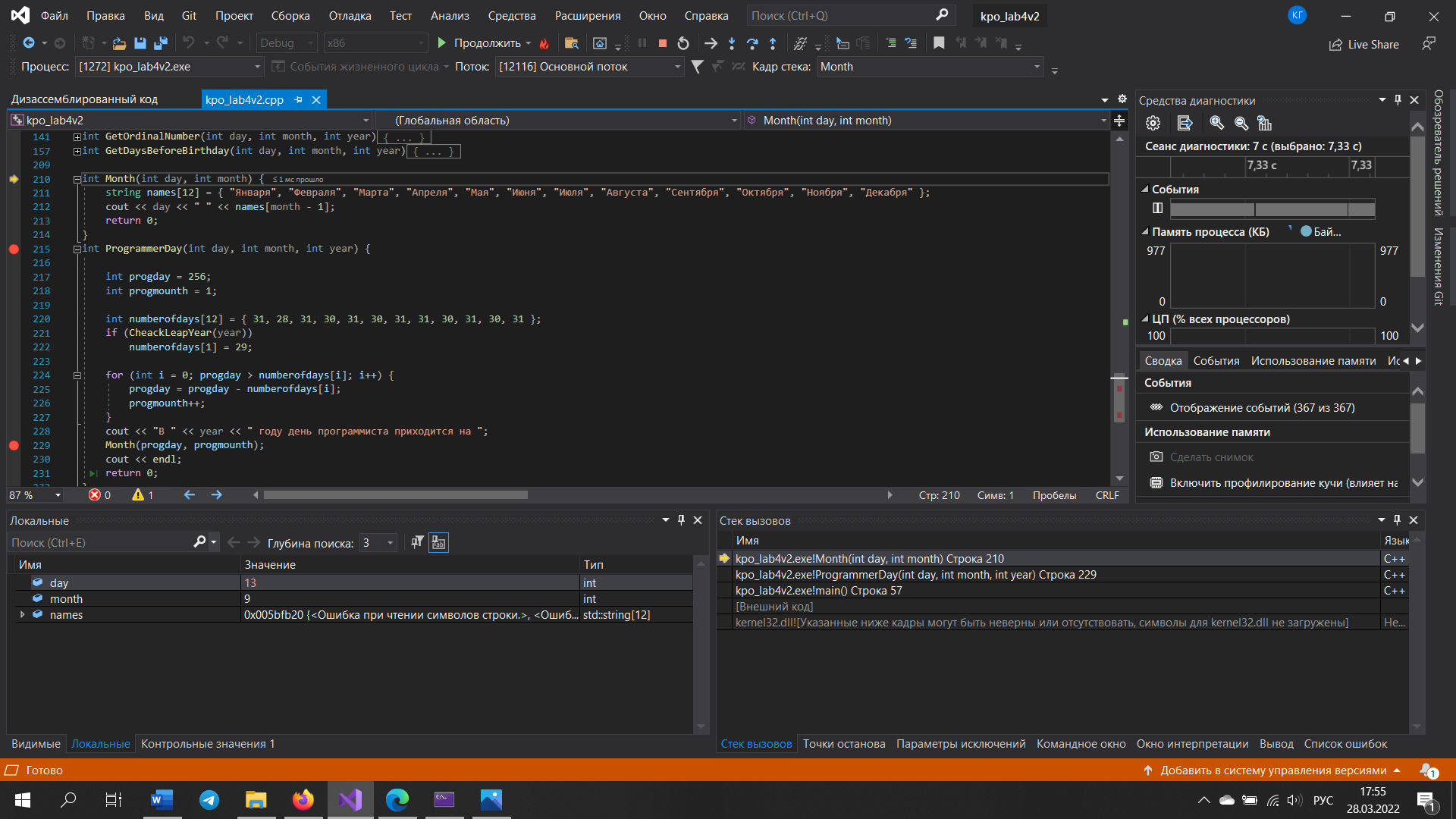


4.

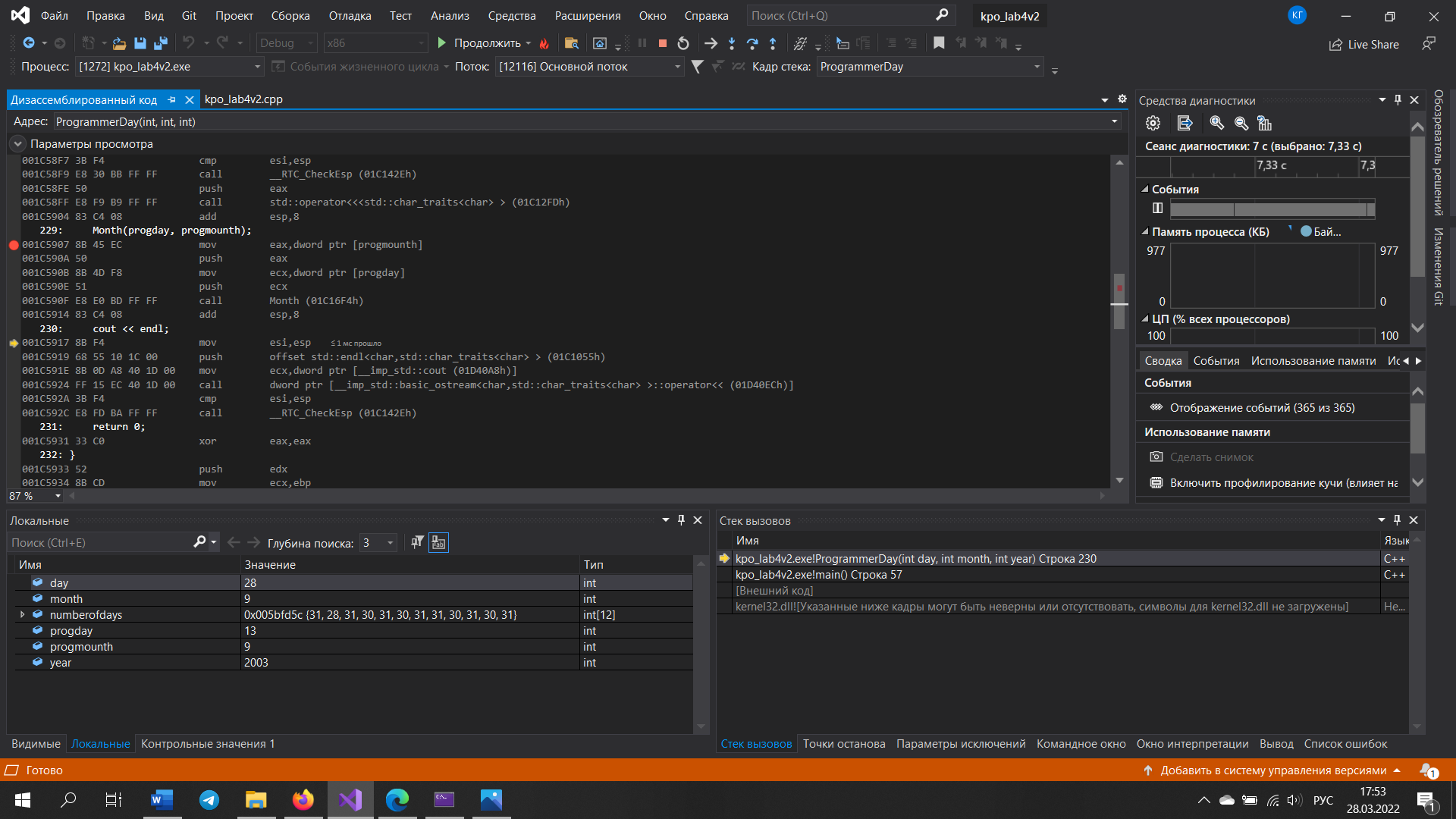


Задание 3

Исследуйте стек вызовов функций в режиме отладки. Во время шага с заходом в функцию Month она появилась на вершине стека:



После выполнения функции Month стек стал выглядеть следующим образом:

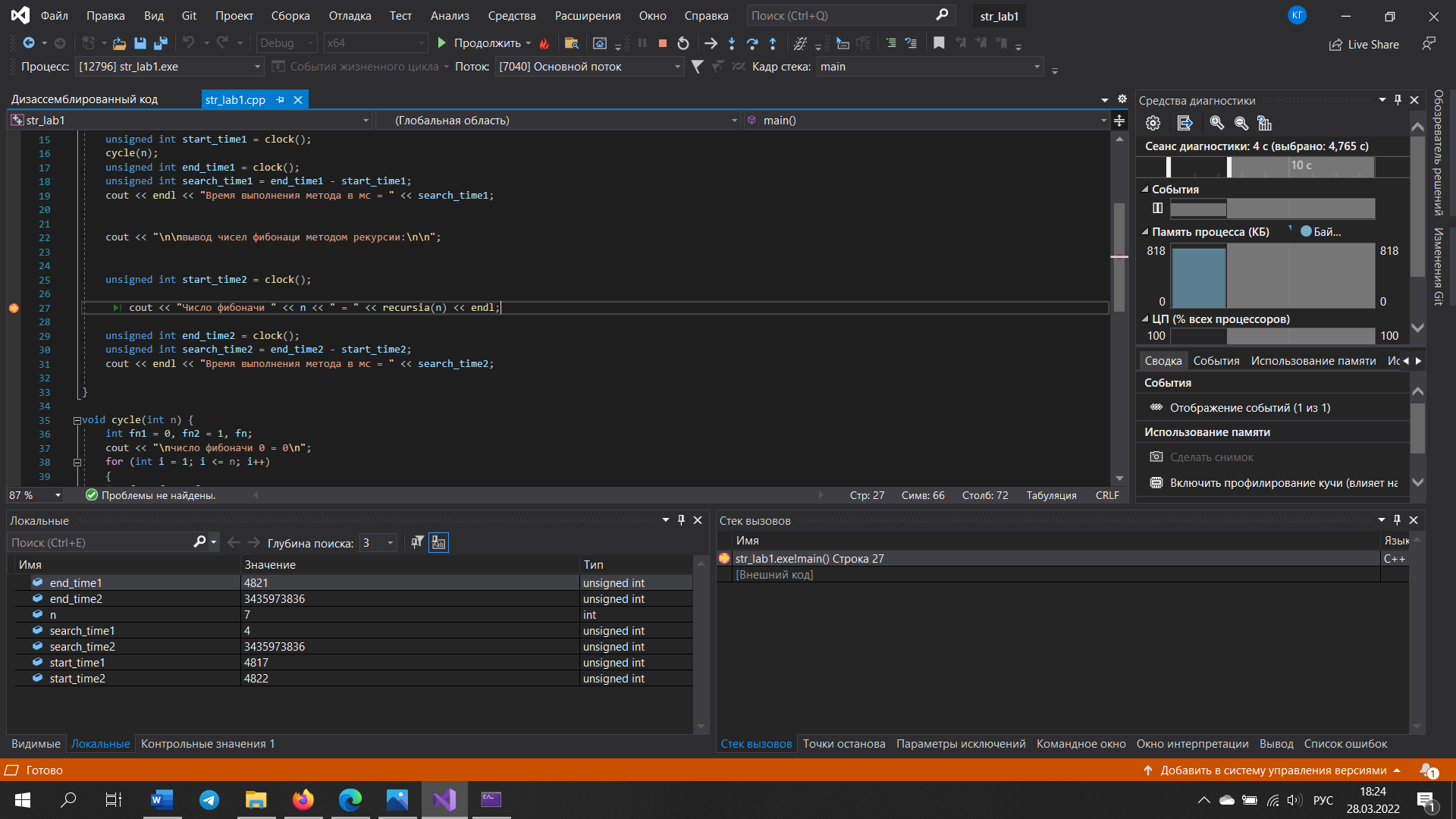


Задание 5 (дополнительное)

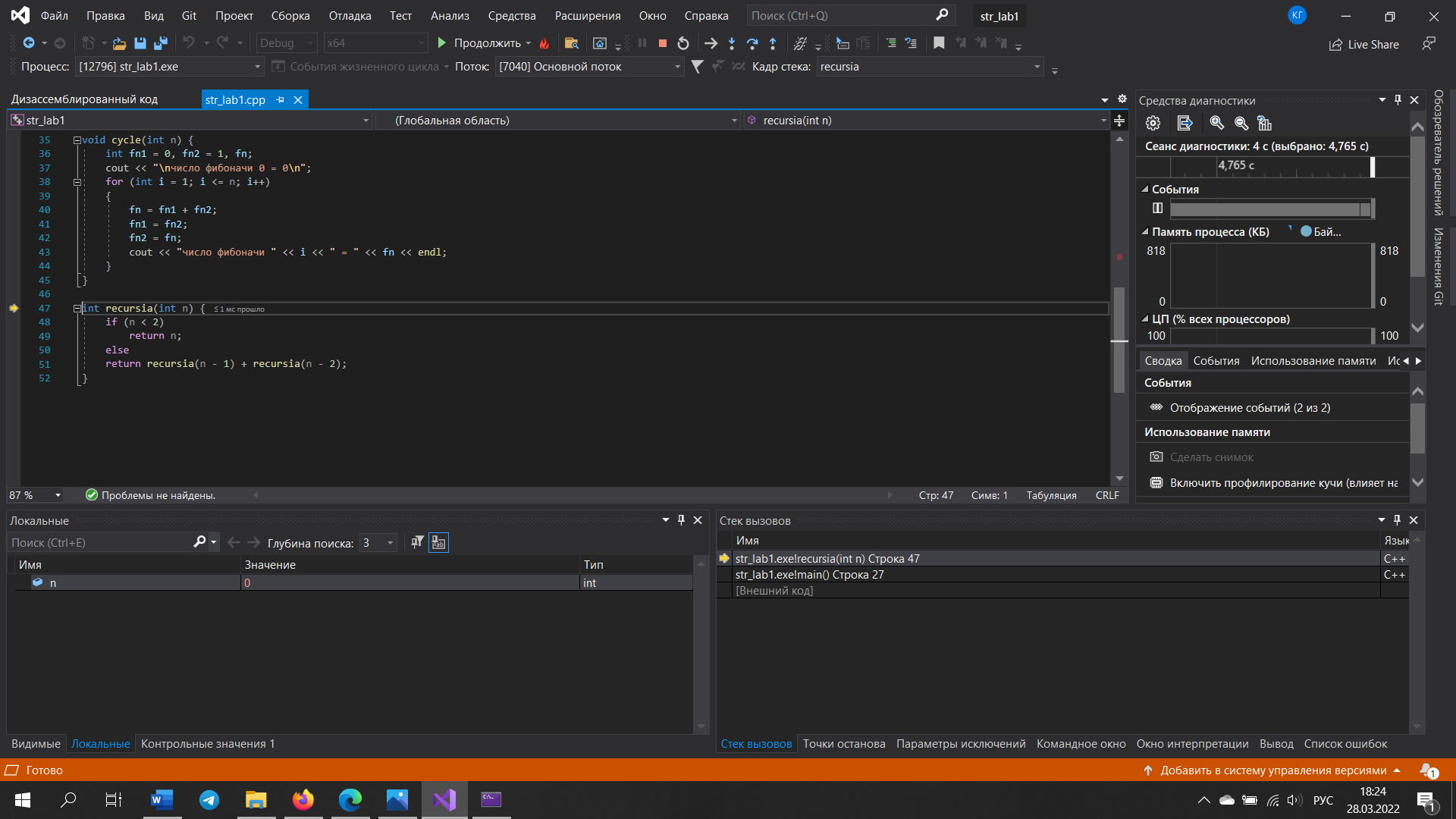
Выполнить пункты 2-4 для лабораторной работы по дисциплине «Основы алгоритмизации и программирования», желательно на тему «Рекурсия»

Как пример взята 1 лабораторная по предмету «Алгоритмы и структуры данных»

1. До захода в функцию:

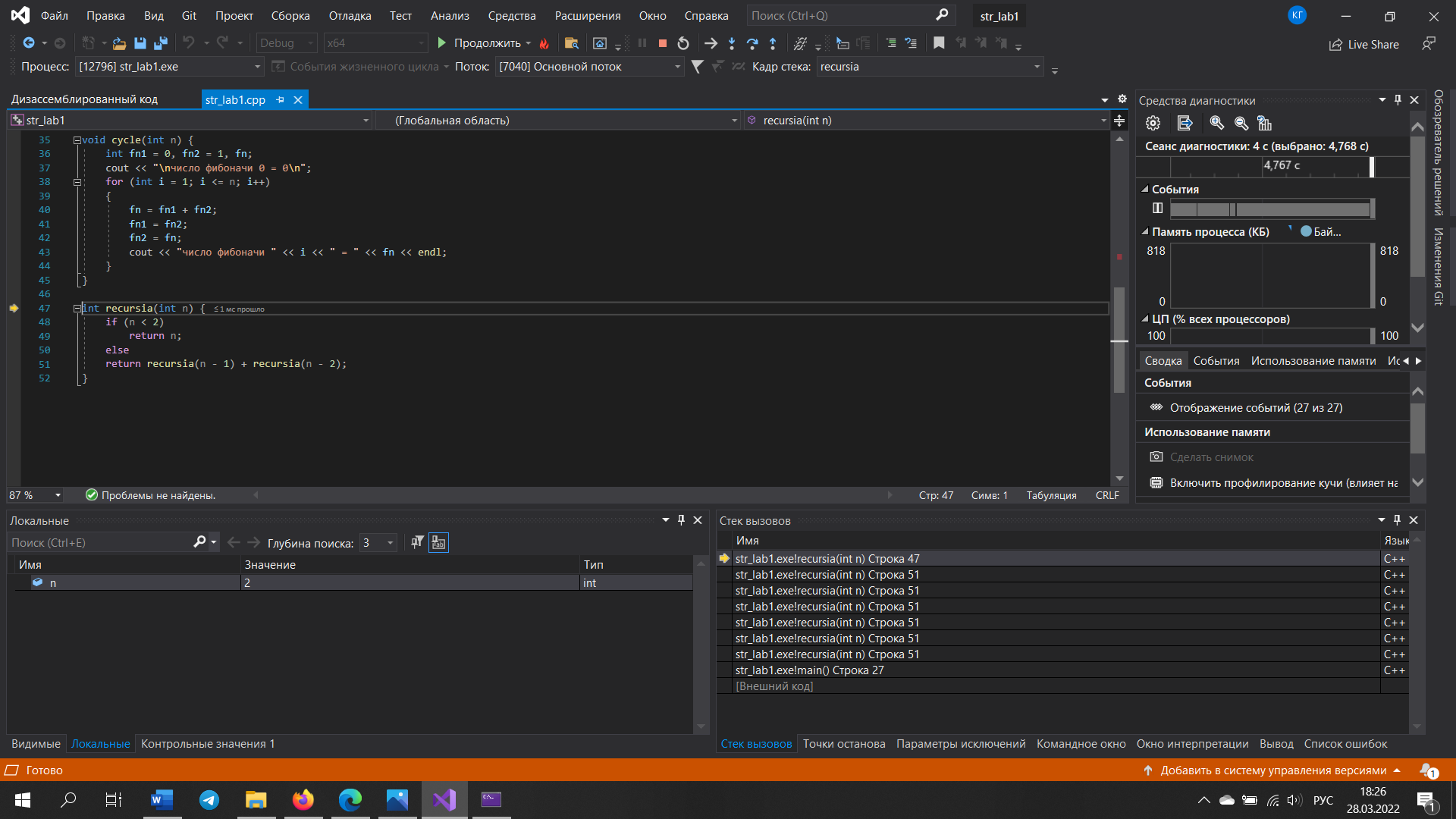


1. Выполняем шаг с заходом в функцию:

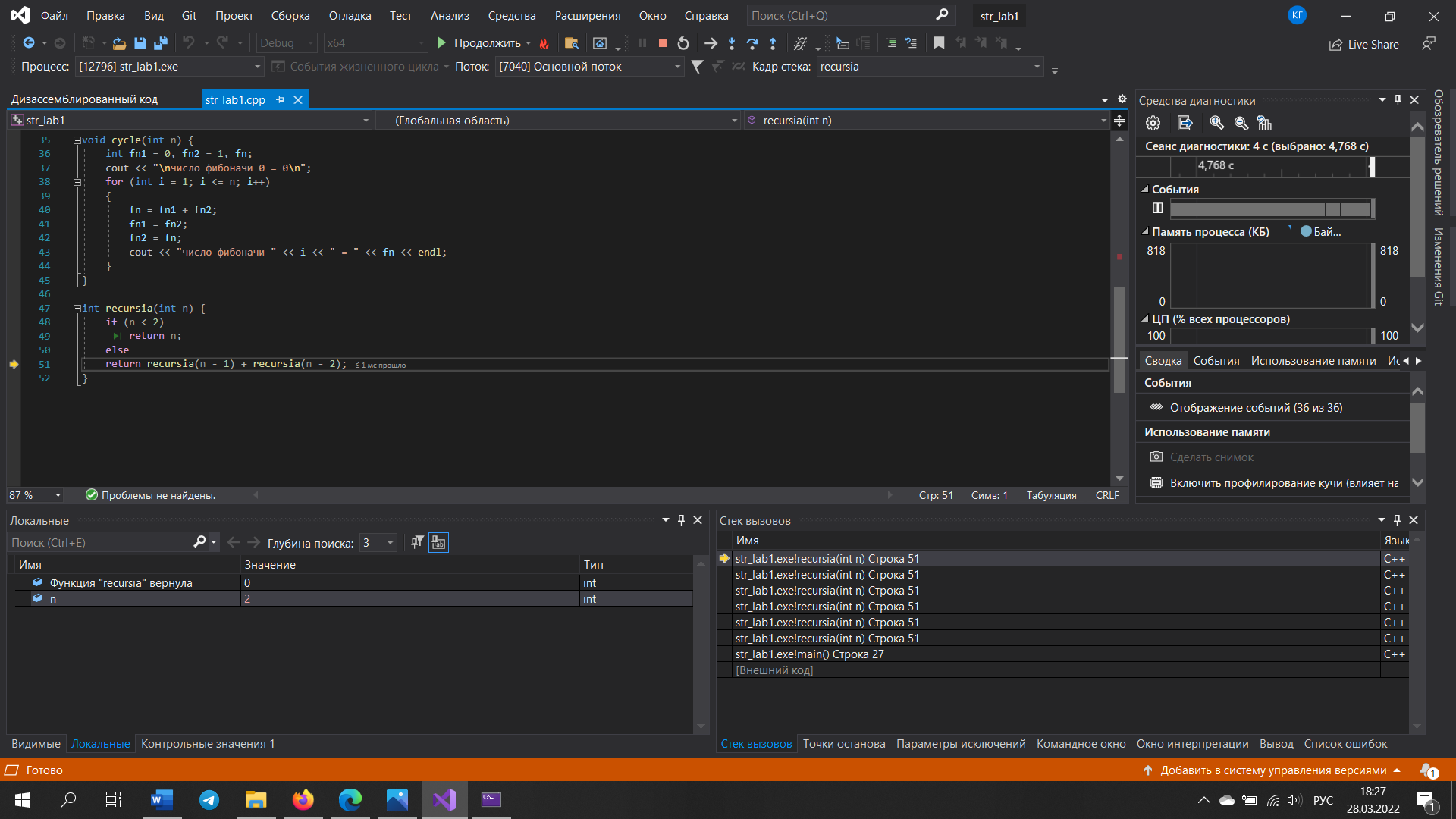


Мы переместились к реализации функции. В стеке появилась новая функция.

1. Рекурсия в стеке



1. Освобождение стека. Функция рекурсии вернула 0 и в стеке стало на 1 функцию меньше.



1. Выход из функции рекурсии. Стек освобождён от функций рекурсии.

