Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Факультет прикладной математики и информатики

Отчёт по лабораторной работе

Работа с массивами случайных чисел

Выполнил:

Студент 3821Б1ПМ3

Жеребцов В.А.

Проверил:

Заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений, младший научный сотрудник

Лебедев И.Г.

Нижний Новгород

2021 г.

Содержание

[Введение 3](#_Toc270962758)

[Постановка задачи 4](#_Toc270962759)

[Руководство пользователя 5](#_Toc270962760)

[Руководство программиста 6](#_Toc270962761)

[Описание структур данных 6](#_Toc270962762)

[Описание алгоритмов 8](#_Toc270962763)

[Описание структуры программы 8](#_Toc270962764)

[Эксперименты 9](#_Toc270962765)

[Заключение 11](#_Toc270962765)

[Литература 12](#_Toc270962766)

[Приложения 13](#_Toc270962767)

[Приложение 1 13](#_Toc270962768)

# Введение

Программирование — это интересный, полезный и увлекательный процесс, благодаря которому создаются программы – набор инструкций, которые приводятся в исполнение компьютерами.

Одной из ключевых задач компьютера является работа с данными. В том числе и со случайно генерируемыми наборами данных, о которых пойдет речь в настоящей работе.

Случайные числа — это одна из основных составляющих любого языка программирования, на них строятся многие алгоритмы. Они имеют применение в физике, например, в исследованиях электронного шума, в инженерном деле и исследовании операций. Многие методы статистического анализа требуют использования случайных чисел.

В ходе выполнения лабораторный работы на языке программирования «С» будет написана программа, работающая со случайными числами.

# Постановка задачи

Программа генерирует множество случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся с клавиатуры. После чего подсчитывает выводит сумму, которая получается следующим образом: все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются.

# Руководство пользователя

После запуска программы пользователь видит сообщение «enter elements». Здесь нужно ввести значение количества элементов, которые нужно получить (рис. 1).

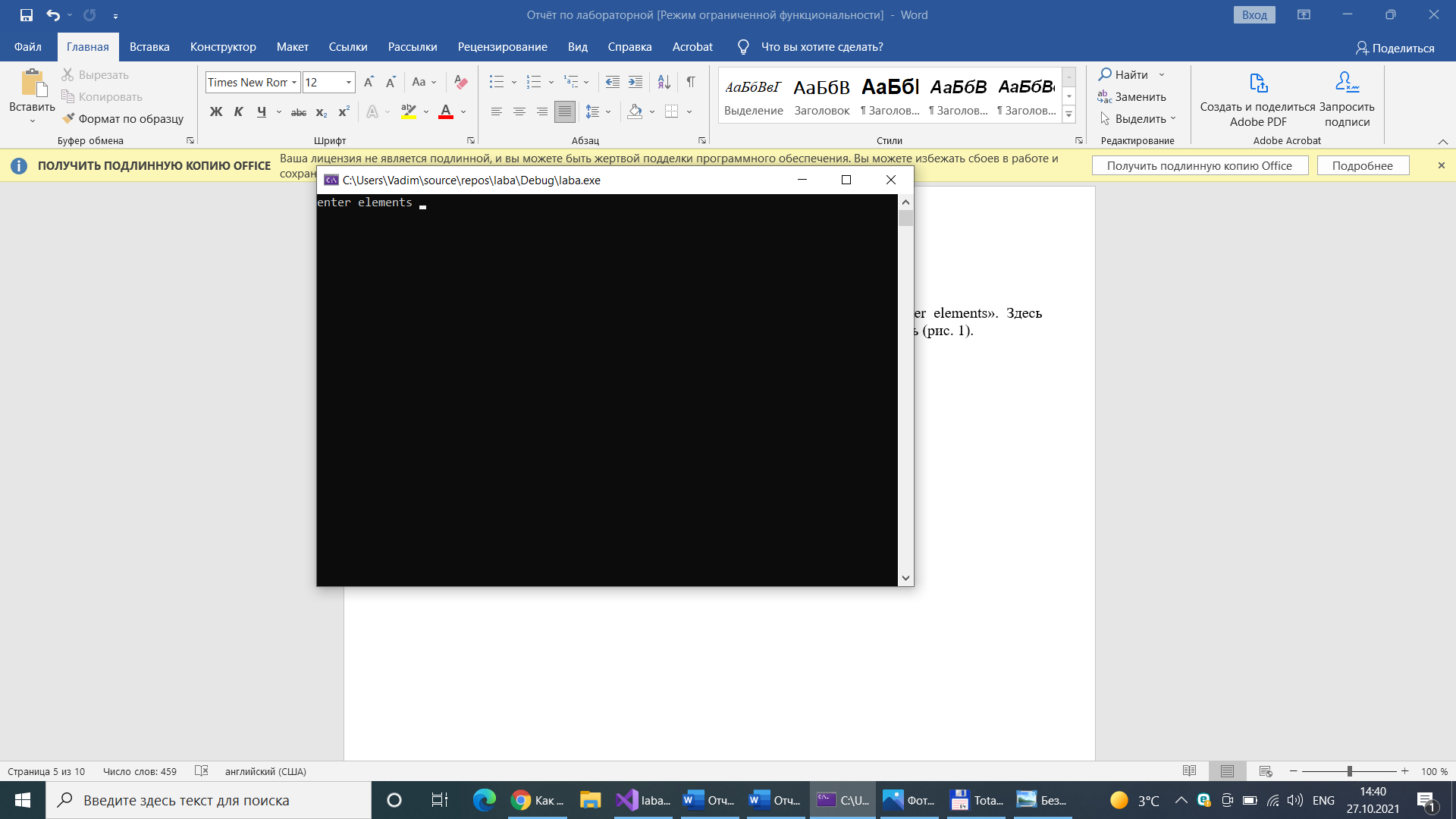


Рисунок 1

Далее пользователь должен ввести минимальное и максимальное значения диапазона: сначала ввести минимальное (рис. 2), затем нажать «Enter» и ввести максимальное (рис. 3).

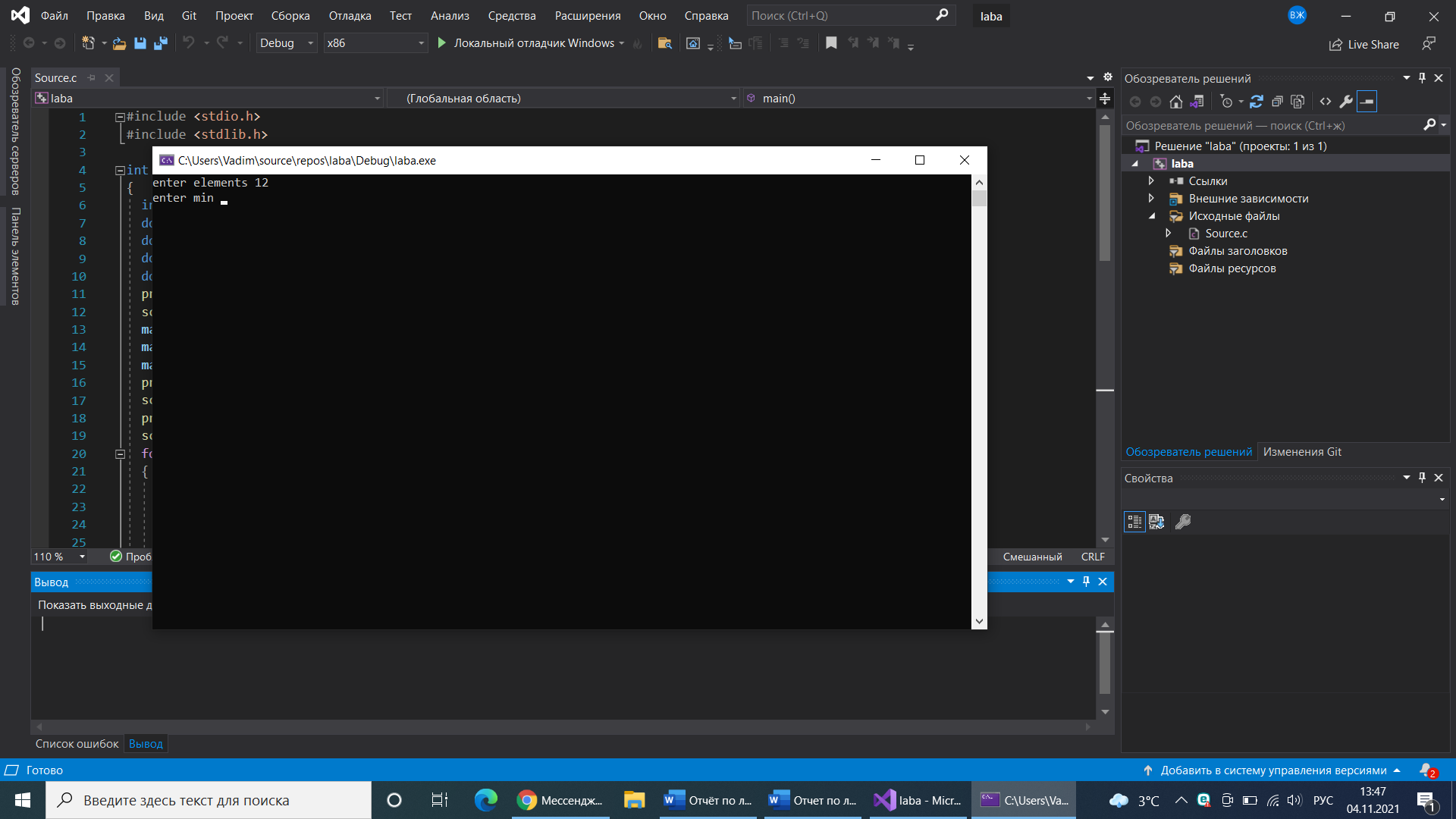


Рисунок 2

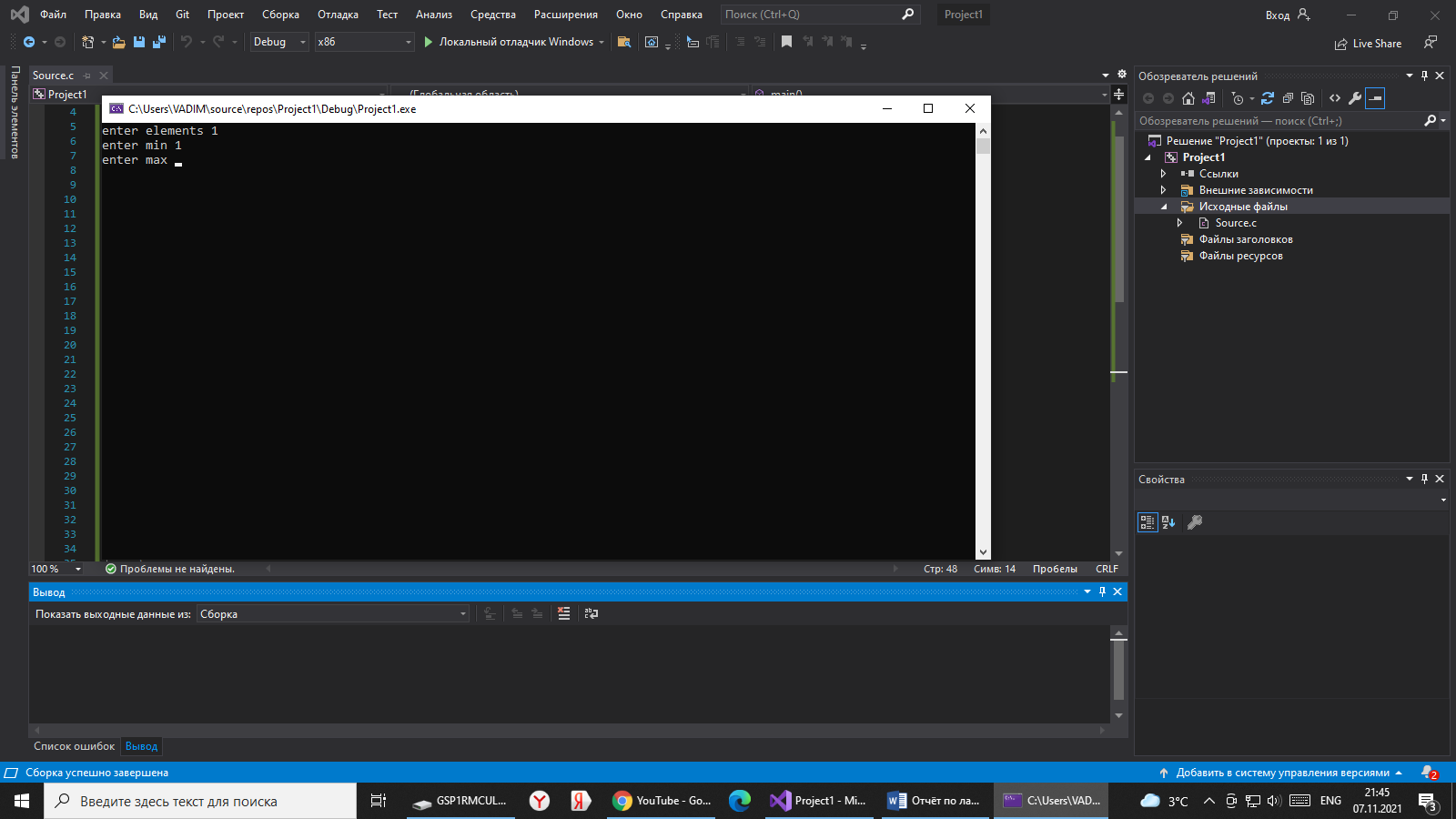


Рисунок 3

Например, введем количество элементов равное 20, начинающихся от 1 (минимальное значение), заканчивающихся 100 (максимальное значение) (рис. 4):

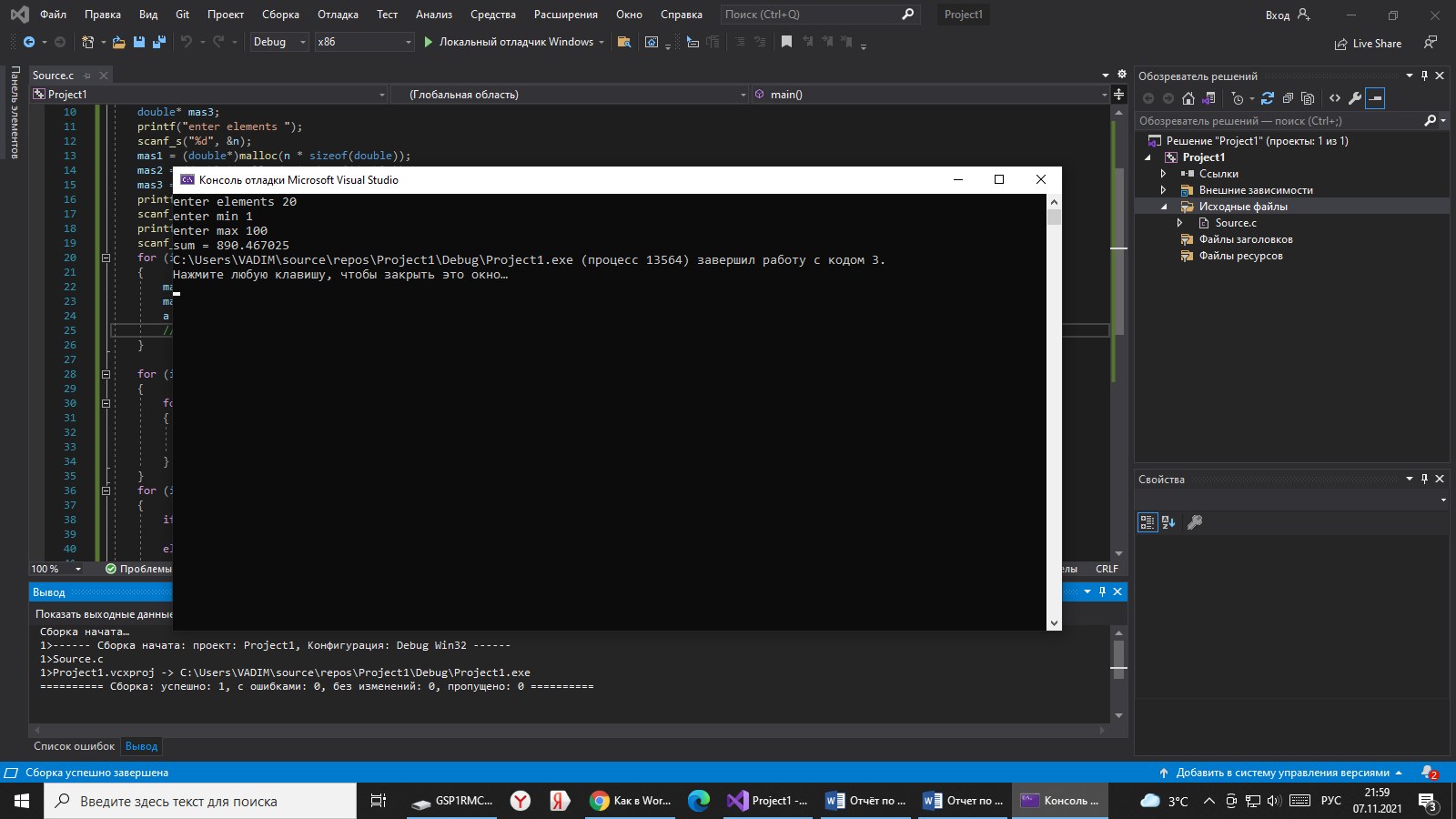


Рисунок 4

После того как пользователь нажмет «Enter», программа выведет результат суммы чисел по правилам данного задания. На этом программа завершится (рис. 5).

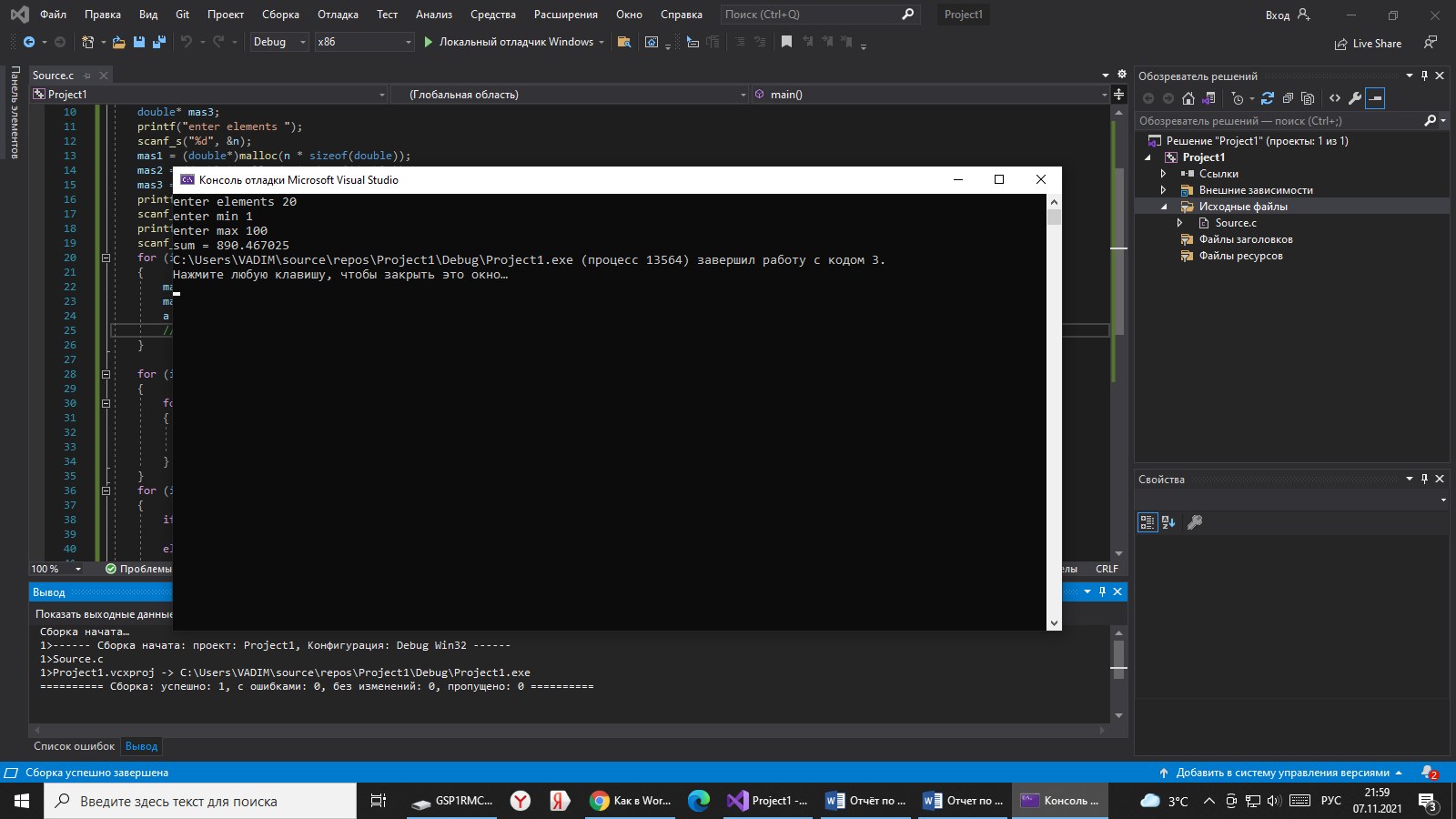


Рисунок 5

**Руководство программиста**

## Описание структуры программы

1. Подключение нужных библиотек.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h> |

1. Объявление функции main(), получение входных данных от пользователя, создание двух динамических массивов. Первого – массива masNum типа double и размера n (n – число сгенерированных чисел) и второго – массива masSum того же типа и размера, что первый. Прохождение по массиву от 1 до n, присвоение значения номера каждого элемента в массив masNum, генерирование случайных чисел в нужном диапазоне и нужного размера и присвоение этих чисел в массив masSum, присвоение переменной a дробной части каждого числа и вывод на экран сгенерированных чисел, их номера и дробной части.

|  |
| --- |
| int main()  {  int n = 0, min = 0, max = 0, i = 0, j = 0;  double ran = 0.0, a = 0.0, sum = 0.0;  double\* masNum;  double\* masSum;  printf("enter elements ");  scanf\_s("%d", &n);  printf("enter min ");  scanf\_s("%d", &min);  printf("enter max ");  scanf\_s("%d", &max);  masNum = (double\*)malloc(n \* sizeof(double));  masSum = (double\*)malloc(n \* sizeof(double));  for (i = 1; i <= n; i++)  {  masSum[i] = (double)rand() / RAND\_MAX \* ((double)max - (double)min) + (double)min;  a = ((double)masSum[i] - (int)masSum[i]) \* 1000000;  } |

1. Прохождение по массиву от 1 до n, сравнение элементов из массива masSum с элементами из данного массива: если они совпали, то зануляем элементы из массива masNum.

|  |
| --- |
| for (i = 1; i <= n; i++)  {  for (j = 1; j <= n; j++)  {  if ((int)masNum[j] == i)  masNum[j] = 0;  }  } |

1. Прохождение по массиву от 1 до n, сравнение элементов из массива masNum с нулём: если они совпали, то из переменной sum вычитаем значение элемнты из массива masNum, если не совпали, то прибавляем.

|  |
| --- |
| for (i = 1; i <= n; i++)  {  if (masNum[i] == 0)  sum = sum - masSum[i];  else  sum = sum + masSum[i];  } |

1. Вывод на экран значения суммы. Очистка памяти.

|  |
| --- |
| printf("sum = %lf", sum);  free(masSum);  free(masNum);  return 0;  } |

## Описание структур данных

* Для работоспособности программы используются библиотеки: stdio.h и stdlib.h.
* В программе используются динамические массивы masNum и masSum типа double.
* Переменные min, max, a, ran, sum используются в программе с типом double.
* Переменные i, j, n используются с типом int.

## Описание алгоритмов

1. *Алгоритм выделения дробной части числа*. Дробную часть получаем путем вычитания из числа типа double это же число типа int и умножаем на 1000000, так как по умолчанию количество знаков после запятой равняется 6.
2. *Алгоритм заполнения массива случайными числами.* (рис. 12).

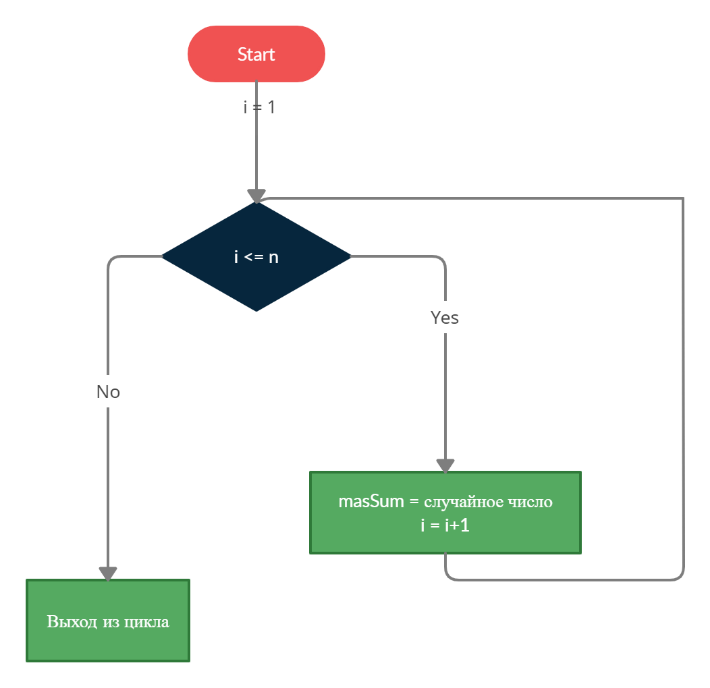
**

Рисунок 12

1. *Алгоритм суммирования.* Сначала проходим по одному массиву и зануляем его элементы, значения которых совпадает с условием, затем идём по другому массиву и проверяем, совпадают ли значения его элементов с нулём: если нет, то прибавляем это значение к переменной sum, если да, то вычитаем.

# Эксперименты

1. Проверим работоспособность программы на небольших значениях: количество элементов равно 10, диапазон равен от 1 до 15

* Сжатый вариант вывода (рис. 6)

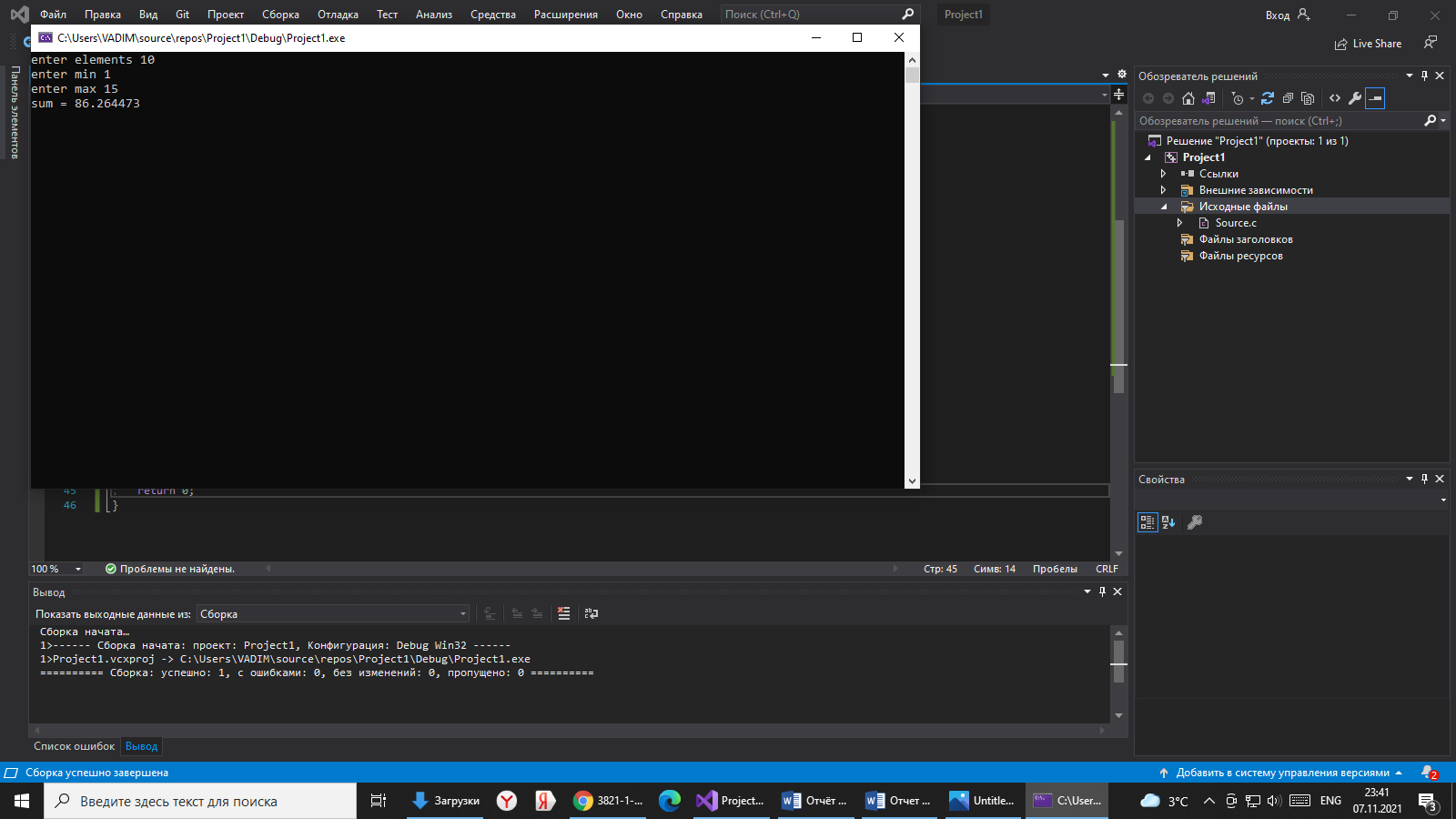


Рисунок 6

* Развернутый вариант вывода (рис. 7)

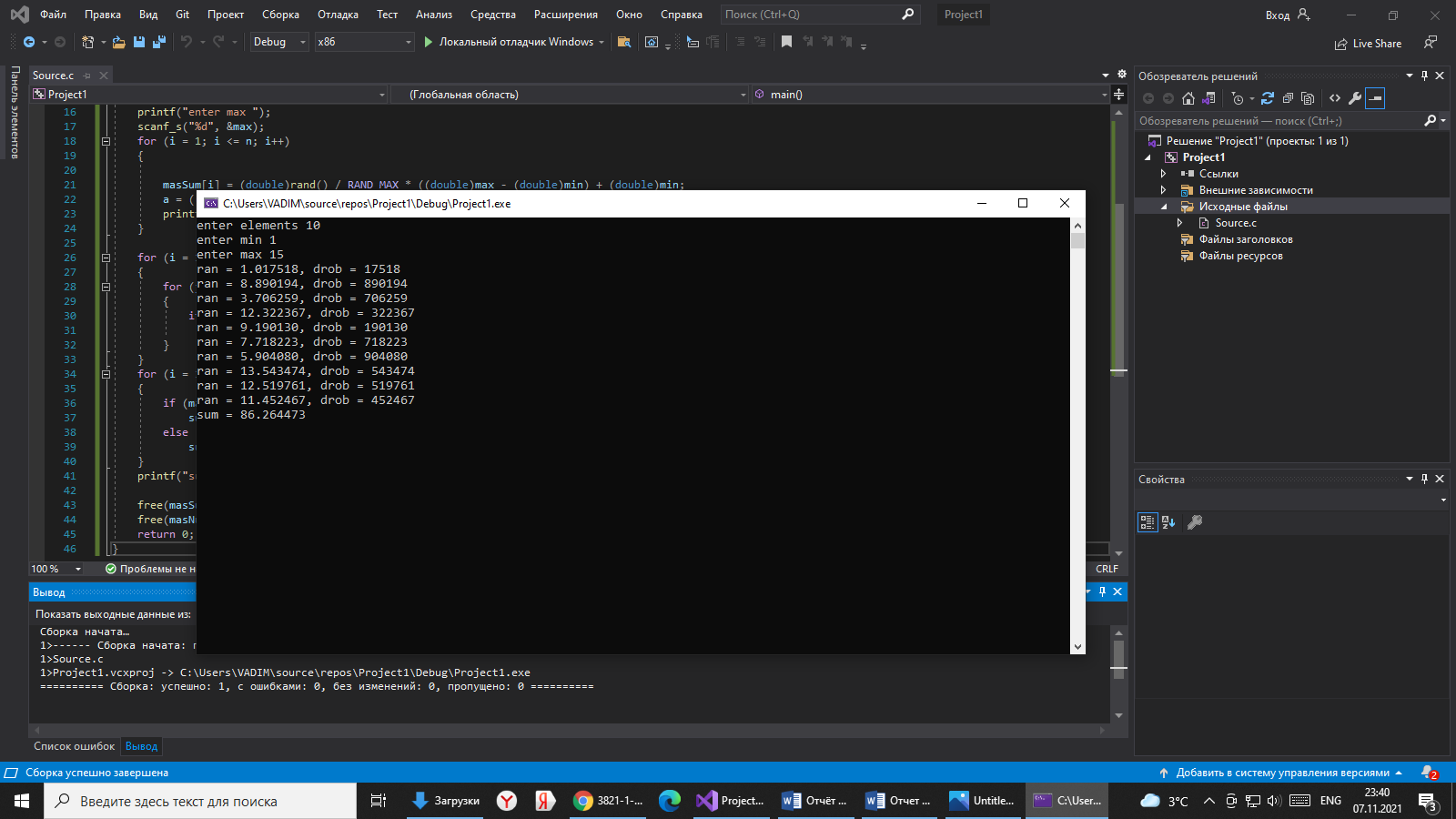


Рисунок 7

1. Введем некоторые различные и значения и проверим работу программы (рис. 8 - 10)

* n = 100, min = 0, max = 599

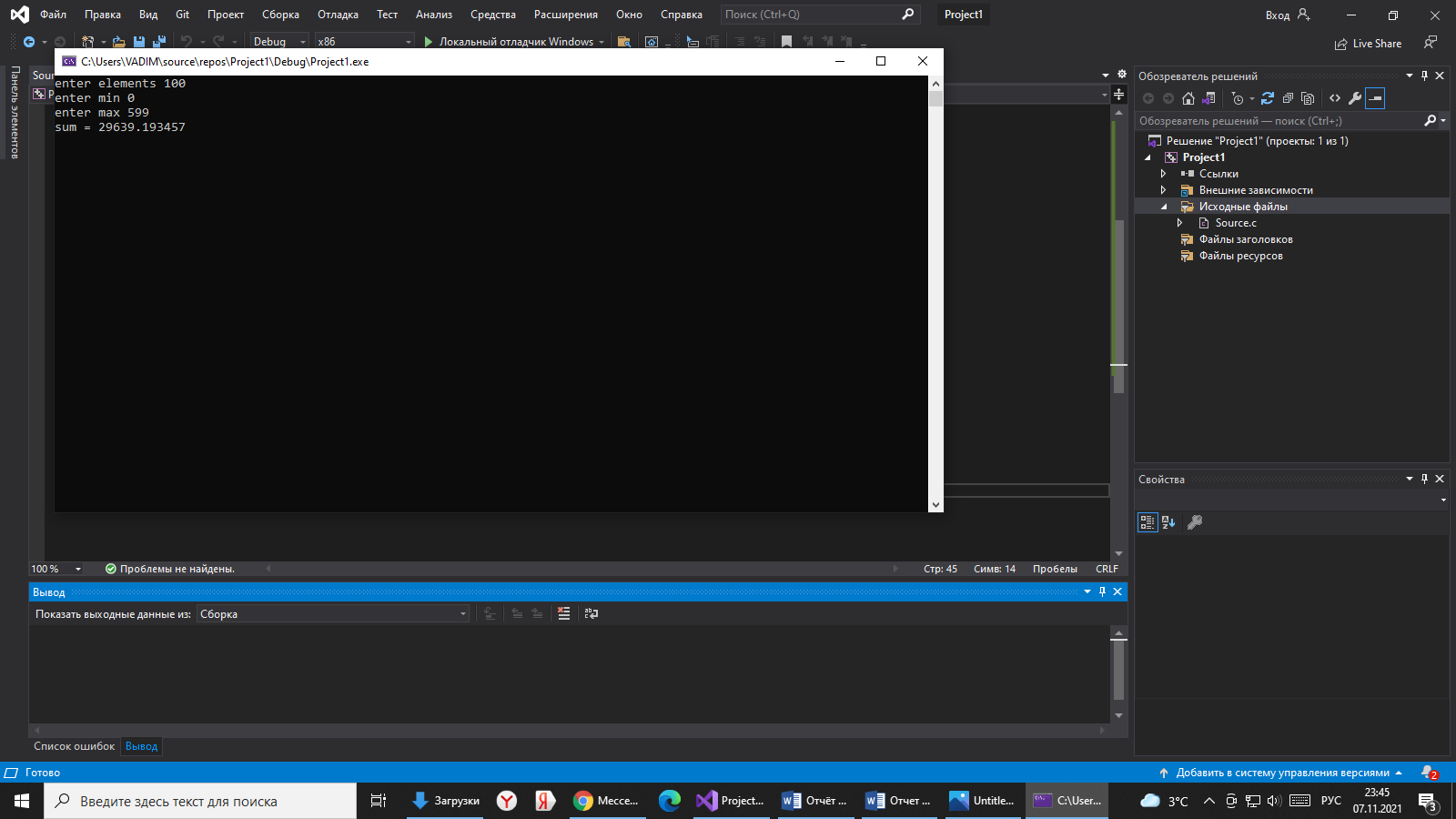


Рисунок 8

* n = 2, min = 10, max = 200

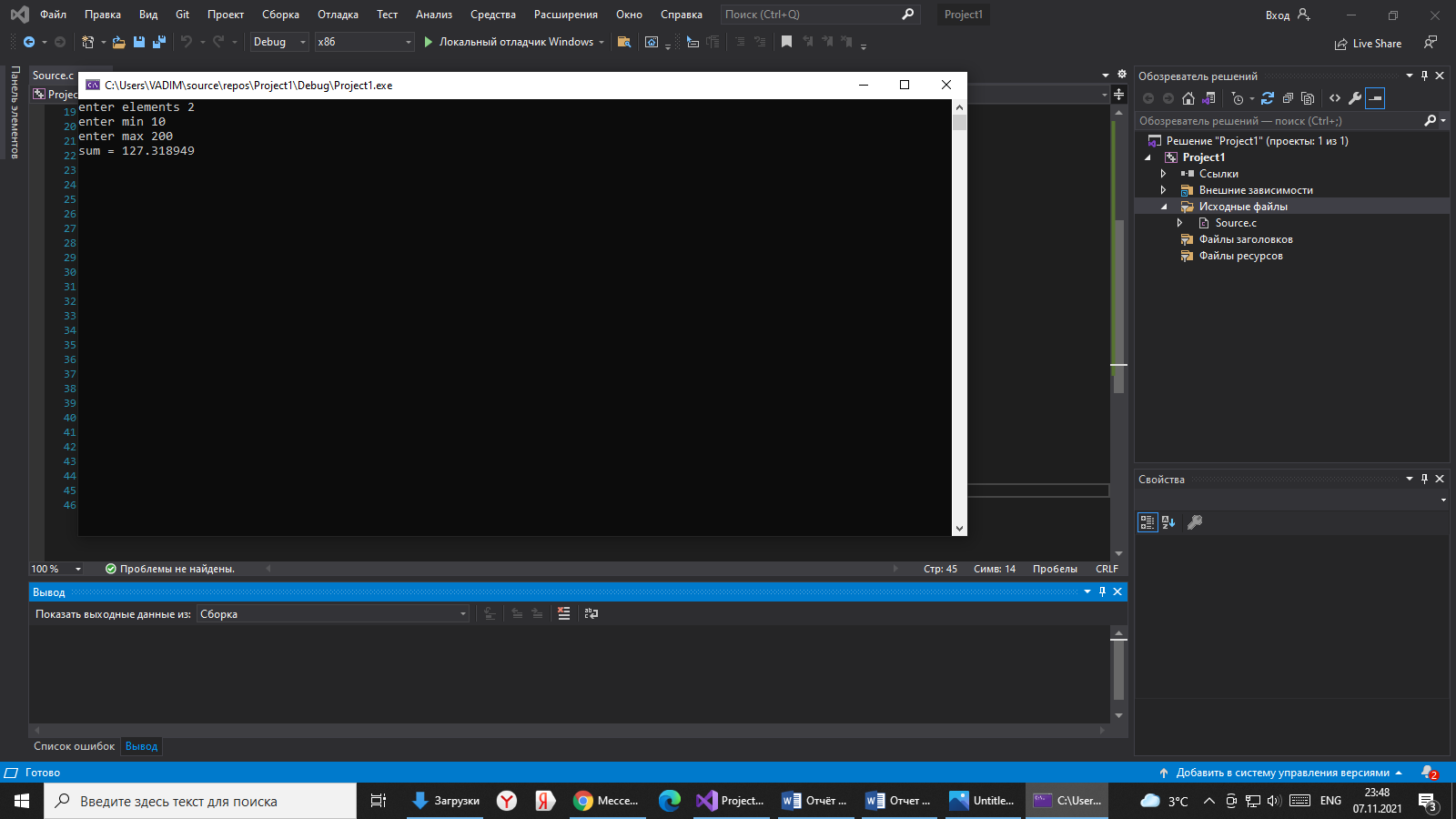


Рисунок 9

* n = 1000, min = 111, max = 3333

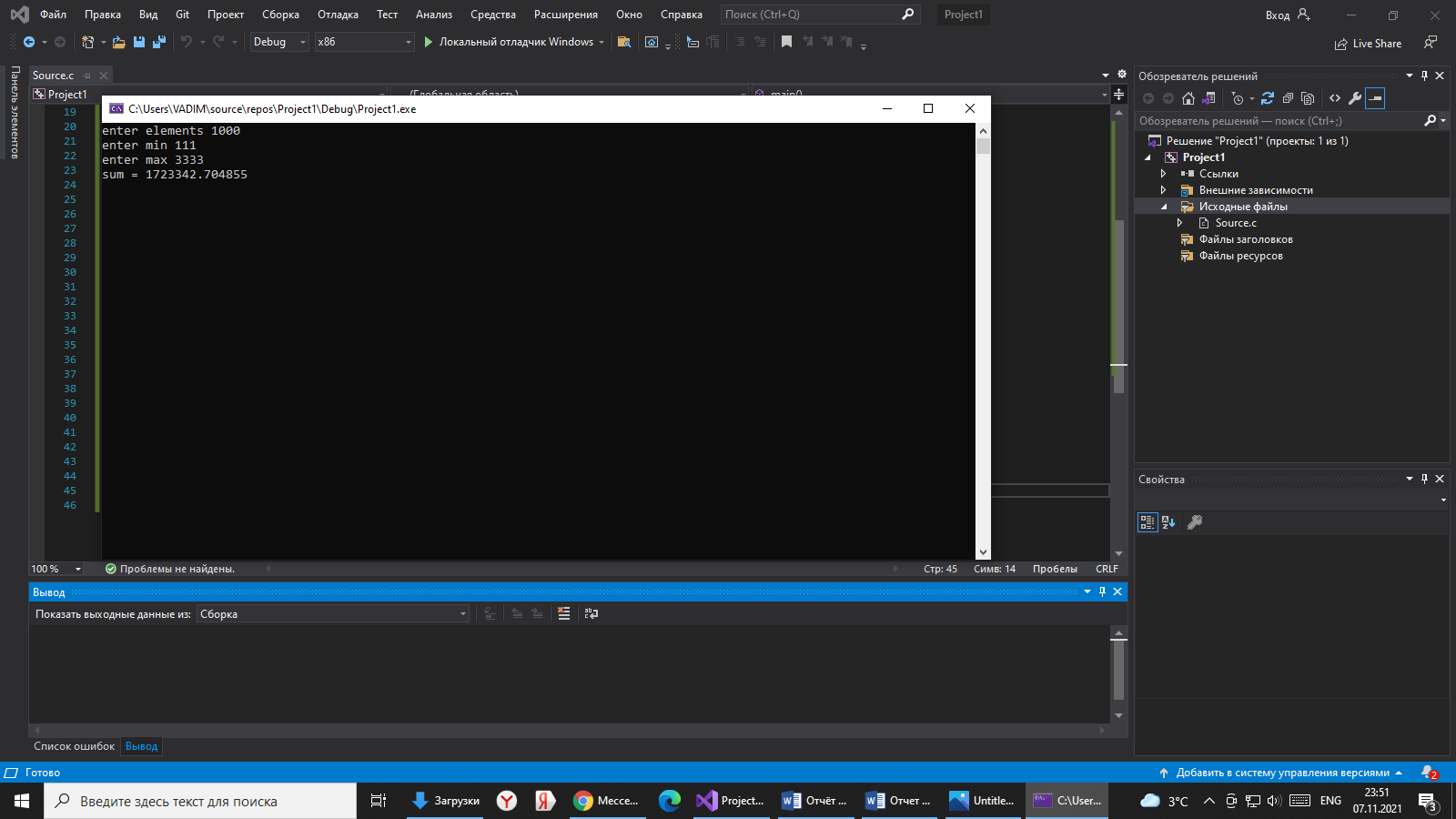


Рисунок 10

1. Проверим работу программы с введенными большими значениями (рис. 11)

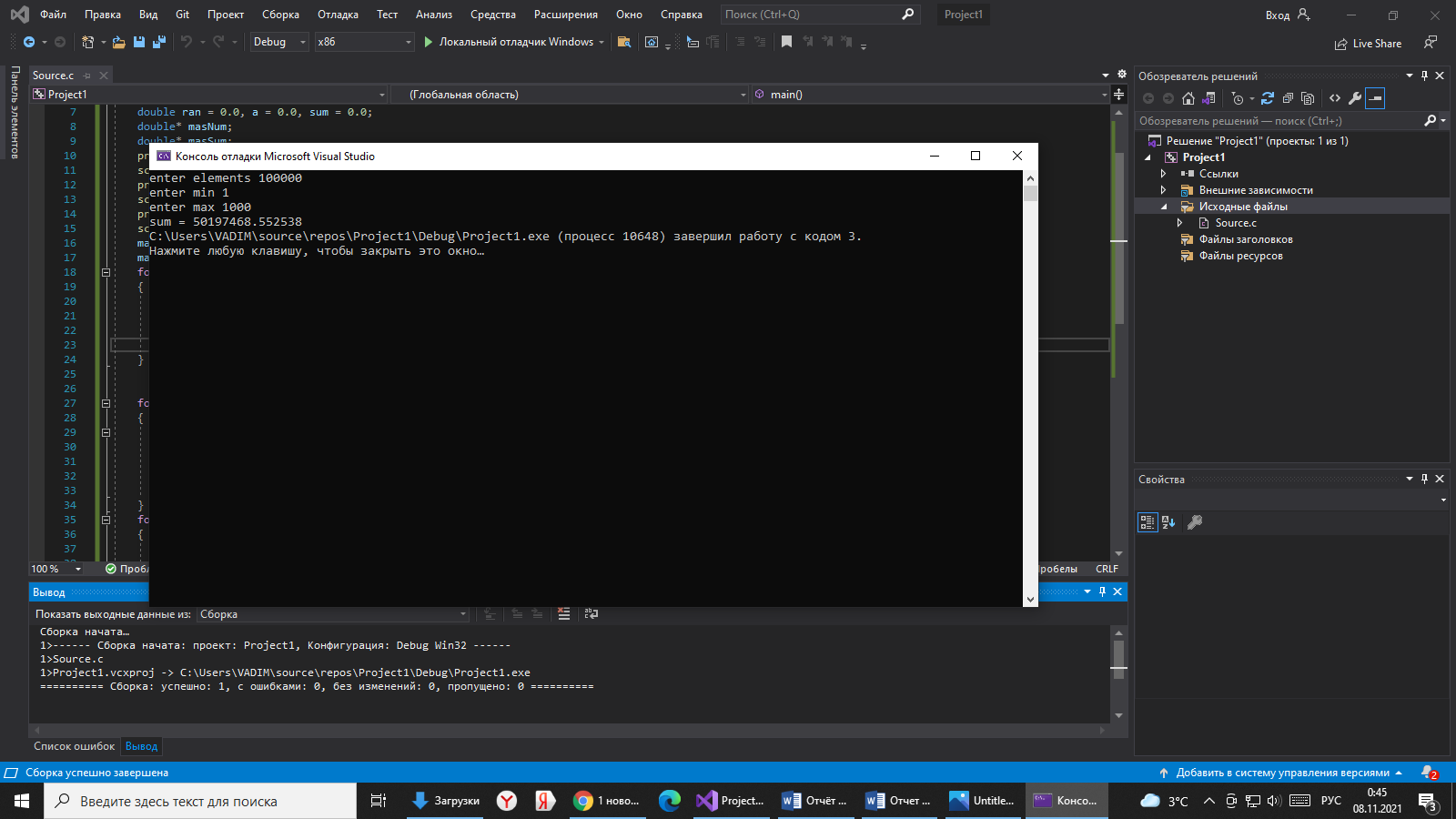


Рисунок 11

# Заключение

Я написал программу на языке «С», которая генерирует множество случайных чисел размера n в диапазоне (min, max), где n, min, max вводятся с клавиатуры. После чего подсчитывает, выводит сумму, которая получается следующим образом: все числа, номера которых совпадают с дробной частью одного из исходных чисел - вычитаются, все остальные прибавляются.

Данной лабораторная работа помогла мне изучить принципы работы с динамическими массивами, различные алгоритмы, принципы работы со случайными числами.

Такие работы позволяют улучшить свои навыки программирования.

# Литература

1. Сборник задач по программированию/Д.М. Златопольский — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 304 с.
2. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. — СПб.: Питер, 2003. —461 с: ил.

# Приложение 1

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n = 0, min = 0, max = 0, i = 0, j = 0;

double ran = 0.0, a = 0.0, sum = 0.0;

double\* masNum;

double\* masSum;

printf("enter elements ");

scanf\_s("%d", &n);

printf("enter min ");

scanf\_s("%d", &min);

printf("enter max ");

scanf\_s("%d", &max);

masNum = (double\*)malloc(n \* sizeof(double));

masSum = (double\*)malloc(n \* sizeof(double));

for (i = 1; i <= n; i++)

{

masSum[i] = (double)rand() / RAND\_MAX \* ((double)max - (double)min) + (double)min;

a = ((double)masSum[i] - (int)masSum[i]) \* 1000000;

}

for (i = 1; i <= n; i++)

{

for (j = 1; j <= n; j++)

{

if ((int)masNum[j] == i)

masNum[j] = 0;

}

}

for (i = 1; i <= n; i++)

{

if (masNum[i] == 0)

sum = sum - masSum[i];

else

sum = sum + masSum[i];

}

printf("sum = %lf", sum);

free(masSum);

free(masNum);

return 0;

}