Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Институт ИВТ

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №1 по теме

«Структурное тестирование»

по дисциплине

«Тестирование программного обеспечения»

Выполнил:

ст-т гр. Б08-191-2 Муллахметов Д.Р.

Принял преподаватель: Старыгина Е.В.

Ижевск 2019

СОДЕРЖАНИЕ

1. Содержание задания3

2. Блок-схема, описание алгоритма4

3. Тестирование базового пути7

3. 1. Потоковый граф7

3. 2. Цикломатическая сложность8

3. 3. Базовое множество независимых линейных путей9

3. 4. Тестовые варианты9

4. Тестирование потоков данных11

4. 1. Информационный граф11

4. 2. Формирование полного набора DU-цепочек12

4. 3. Формирование полного набора отрезков путей в управляющем графе13

4. 4. Построение маршрутов14

5. Области эквивалентности15

6. Контрольный пример16

7. Текст программы18

1. СОДЕРЖАНИЕ ЗАДАНИЯ

Разработать нижеприведенную программу и провести ее структурное тестирование.

Дан массив целых чисел . Найти все пары , такие, что и кратно 2.

2. БЛОК-СХЕМА, ОПИСАНИЕ АЛГОРИТМА

Для начала разработаем алгоритм для решение поставленной задачи. Ниже приведен текст программы на алгоритмическом языке:

Программа

1 считать кол-во чисел;

2 если не число

3 то вывести сообщение об ошибке;

4 конец программы;

5 иначе сохраняем введенное число;

6 конец если;

7 выполнять пока не закончатся числа

7 считать число;

8 если не число

9 то вывести сообщение об ошибке;

4 конец программы;

10 иначе сохраняем введенное число;

11 конец если;

12 конец выполнять;

13 выполнять пока не закончатся пары чисел

14 если первое число равно нулю

15 то если второе число кратно двум

16 то сохраняем пару;

17 конец если;

17 конец если;

18 конец выполнять;

19 если пары есть

20 то вывести пары;

21 иначе сообщить, что пар нет;

22 конец если;

4 конец программы;

Далее представим алгоритм в виде блок-схемы (рис.1).



Рис.1



Продолжение Рис.1

3. ТЕСТИРОВАНИЕ БАЗОВОГО ПУТИ

3. 1. Потоковый граф

Затем блок-схема отображается в потоковый граф, представленный на рис. 2. Видим, что этот потоковый граф имеет восемь регионов.



Рис.2



Продолжение Рис.2

3. 2. Цикломатическая сложность

Определим цикломатическую сложность потокового графа по каждой из трех формул:

1) V(G) = 8 регионов;

2) V(G) = 28 дуг - 22 узла + 2 = 8;

3) V(G) = 7 предикатных узлов + 1 = 8.

3. 3. Цикломатическая сложность

Базовое множество независимых линейных путей:

Путь 1: 1-2-3-4; /ввести для количества чисел не целое число

Путь 2: 1…7-8-9-4;/ввести в набор данных не целое число

Путь 3: 1…8-10…14-17-18-19-21-22-4; /Не вводить нули для пар чисел

Путь 4: 1…8-10…14-17-18-19-20-22-4;/Не ввести ноль хотя бы раз и ввести подходящую пару чисел

Путь 5: 1…8-10…14-15-17-18-19-21-22-4; /Ввести ноль и число не кратное двум

Путь 6: 1…8-10…14-15-17-18-19-20-22-4;/Ввести ноль и не кратное двум число, а также подходящую пару чисел

Путь 7: 1…8-10…19-20-22-4;/Ввести ноль и кратное двум число

3. 4. Тестовые варианты

Тестовый вариант для пути 1 ТВ1:

ИД: кол-во чисел = недопустимое значение

ОЖ.РЕЗ.: вывод сообщения об ошибке

Тестовый вариант для пути 2 ТВ2:

ИД: кол-во чисел = допустимое значение, числа = недопустимое значение

ОЖ.РЕЗ.: вывод сообщения об ошибке

Тестовый вариант для пути 3 ТВ3:

ИД: кол-во чисел = допустимое значение, числа = не равные нулю

ОЖ.РЕЗ.: пары не найдены

Тестовый вариант для пути 4 ТВ4:

ИД: кол-во чисел = допустимое значение, числа = одно число не ноль и пара из нуля и числа, которое кратно двум

ОЖ.РЕЗ.: пары найдены

Тестовый вариант для пути 5 ТВ5:

ИД: кол-во чисел = допустимое значение, числа = ввести ноль и число не кратное двум

ОЖ.РЕЗ.: пары найдены

Тестовый вариант для пути 6 ТВ6:

ИД: кол-во чисел = допустимое значение, числа = ввести ноль и не кратное двум число, а также подходящую пару чисел

ОЖ.РЕЗ.: пары найдены

Тестовый вариант для пути 7 ТВ7:

ИД: кол-во чисел = допустимое значение, числа = ввести ноль и кратное двум число

ОЖ.РЕЗ.: пары найдены

4. ТЕСТИРОВАНИЕ ПОТОКОВ ДАННЫХ

4. 1. Информационный граф

Построим граф программы с управляющими и информационными связями



Рис.3



Продолжение Рис.3

4. 2. Формирование полного набора DU-цепочек

В нашем примере существуют следующие DU-цепочки:

[count, 1,5], [count, 1,7], [number, 7,10], [pair, 16, 20].

4. 3. Формирование полного набора отрезков путей в управляющем графе

Фрагмент отрезка для первого пути



Рис.4

Фрагмент отрезка для второго пути



Рис.5

Фрагмент отрезка для третьего пути



Рис.6

Фрагмент отрезка для четвертого пути



Рис.7

4. 4. Построение маршрутов

Можно построить маршрут, который полностью покрывает набор отрезков путей управляющего графа: 1…8-10…19-20-22-4 (Ввести ноль и кратное двум число).

5. ОБЛАСТИ ЭКВИВАЛЕНТНОСТИ

Для разработанной программы выделим две области эквивалентности: когда нужная нам пара есть, и когда ее нет. Также для входных данных обозначим еще две области: когда ввели менее двух чисел и от двух и более.

Область эквивалентности количества введенных значений



Рис.8

Область эквивалентности значений входных данных



Рис.9

Таблица 1

Области эквивалентности

|  |  |  |  |
| --- | --- | --- | --- |
| **Последовательность** | **Ключевые элементы** | | |
| Один элемент | Нет в последовательности | | |
| Несколько элементов | Нет в последовательности | | |
| Несколько элементов | Первые элементы последовательности | | |
| Несколько элементов | Последний элемент последовательности | | |
| Несколько элементов | Средний элемент последовательности | | |
|  | | | |
| **Входная последовательность** | | **Пара** | **Выходные данные** |
| 0 | | Нет | Пар нет |
| 2 3 4 5 | | Нет | Пар нет |
| 0 2 4 5 6 | | 0 2 | Пара есть |
| 9 8 2 0 4 | | 0 4 | Пара есть |
| 8 5 1 5 0 6 3 5 6 2 | | 0 6 | Пара есть |

6. КОНТРОЛЬНЫЙ ПРИМЕР

Первый тестовый вариант

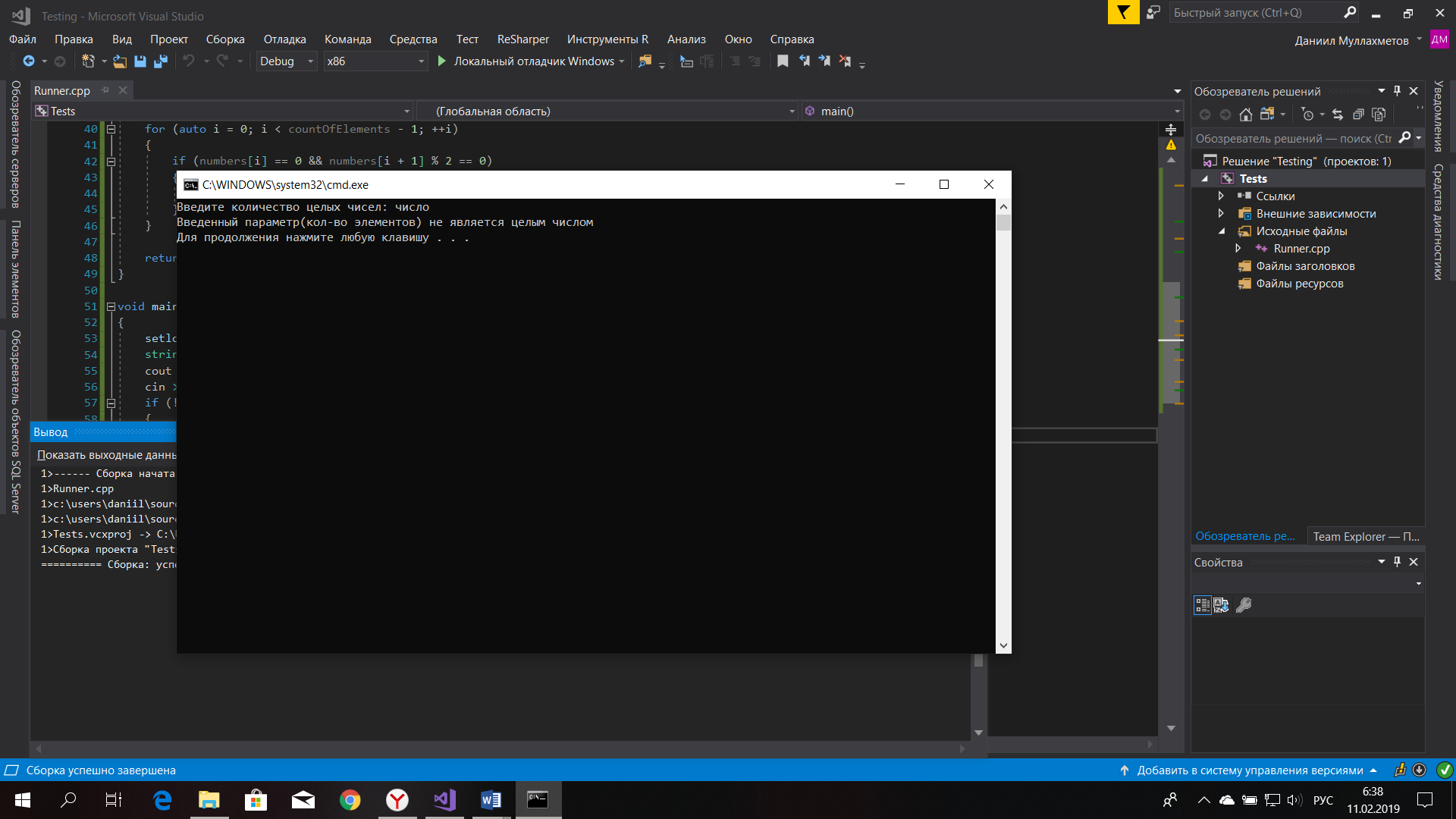


Рис.10

Второй тестовый вариант

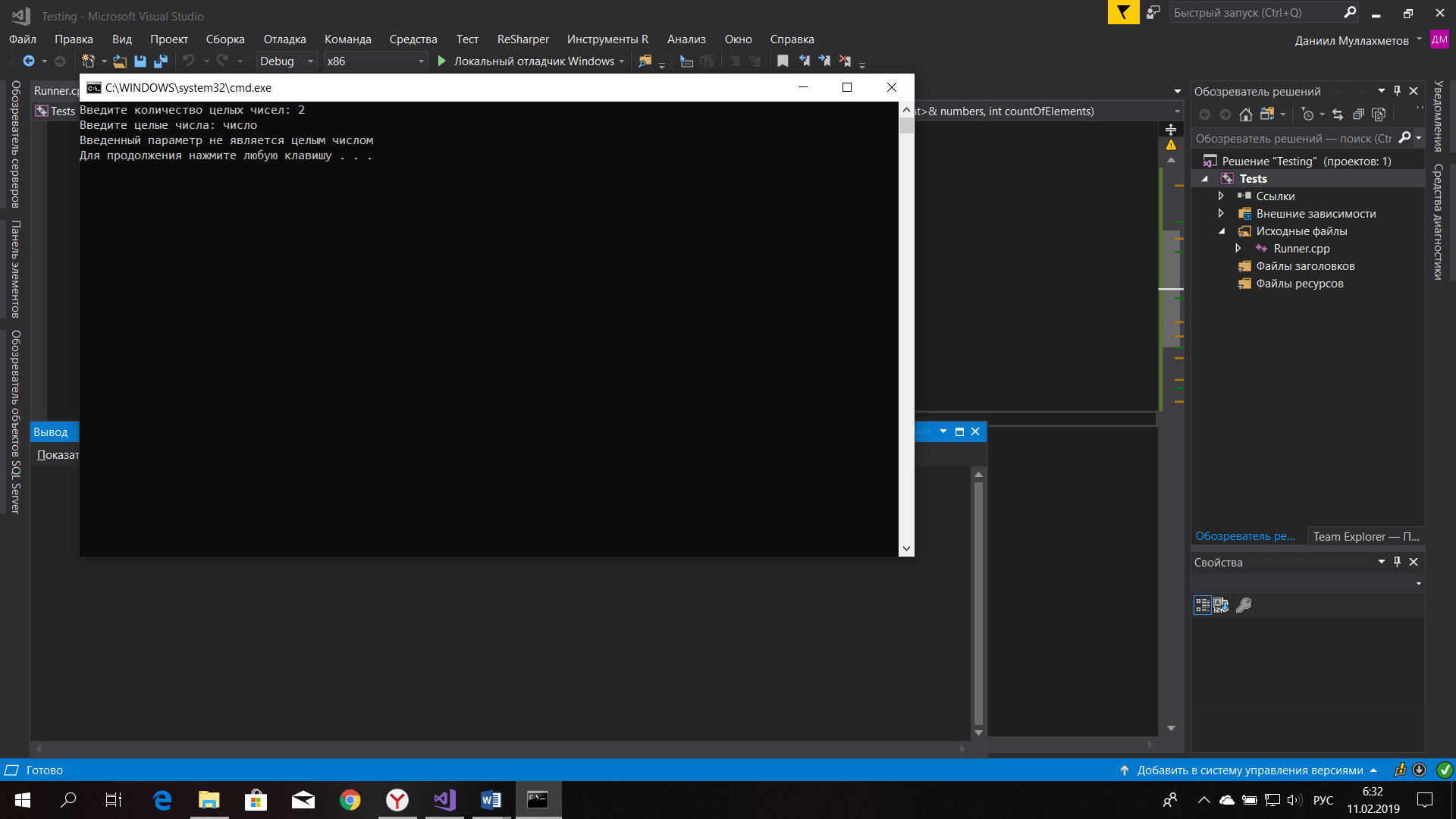


Рис.11

Третий тестовый вариант

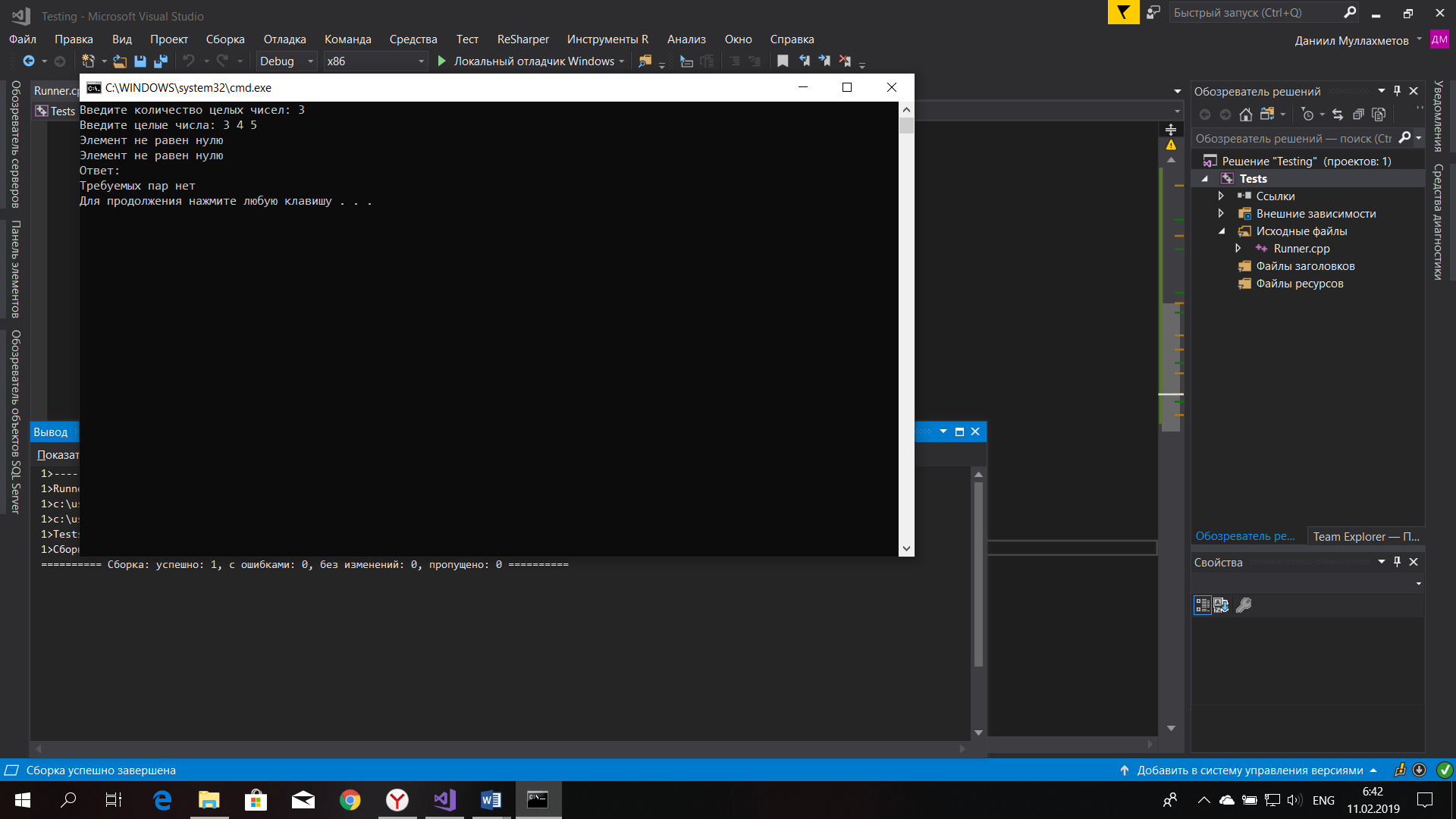


Рис.11

Четвертый тестовый вариант

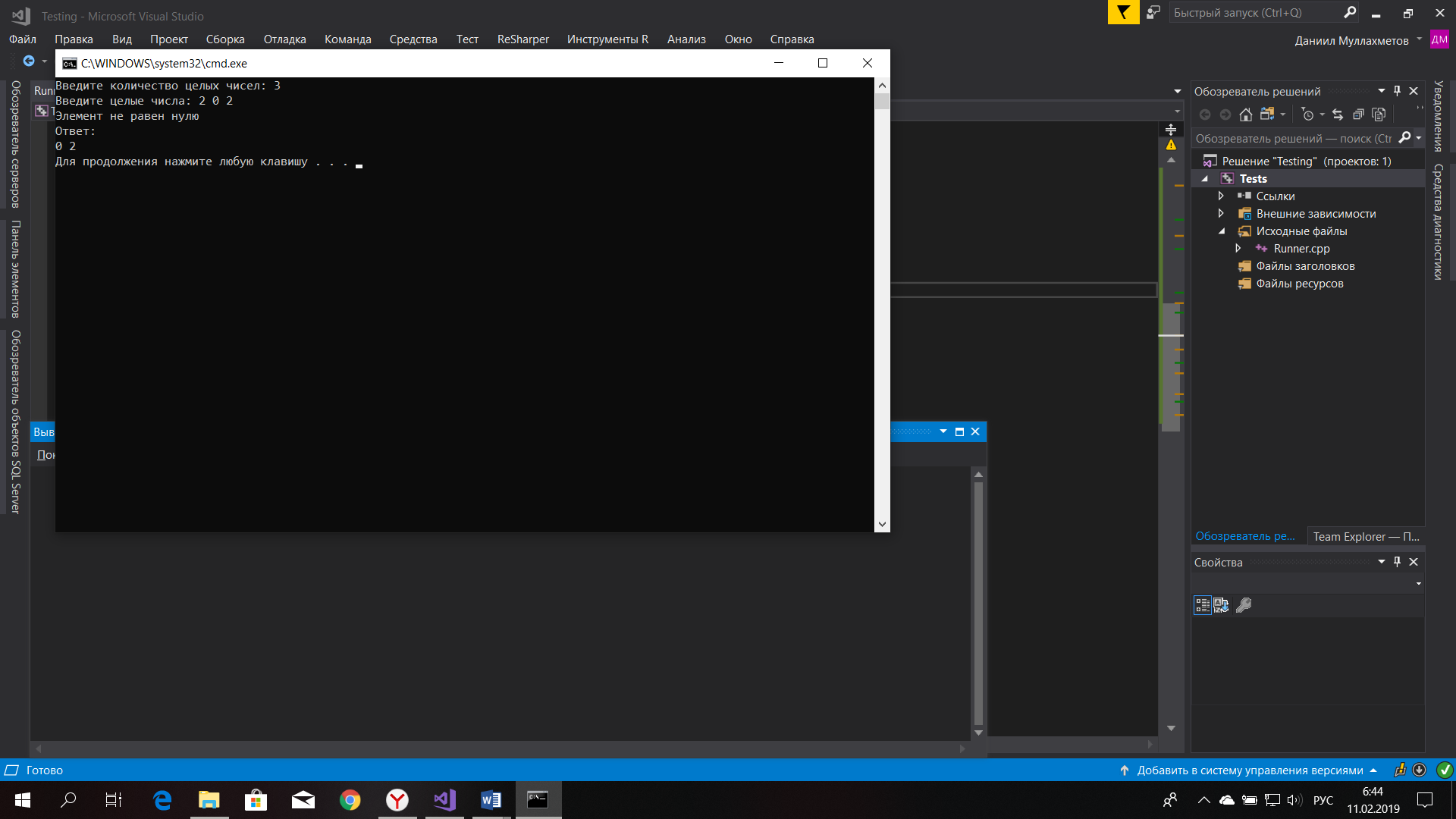


Рис.12

Пятый тестовый вариант

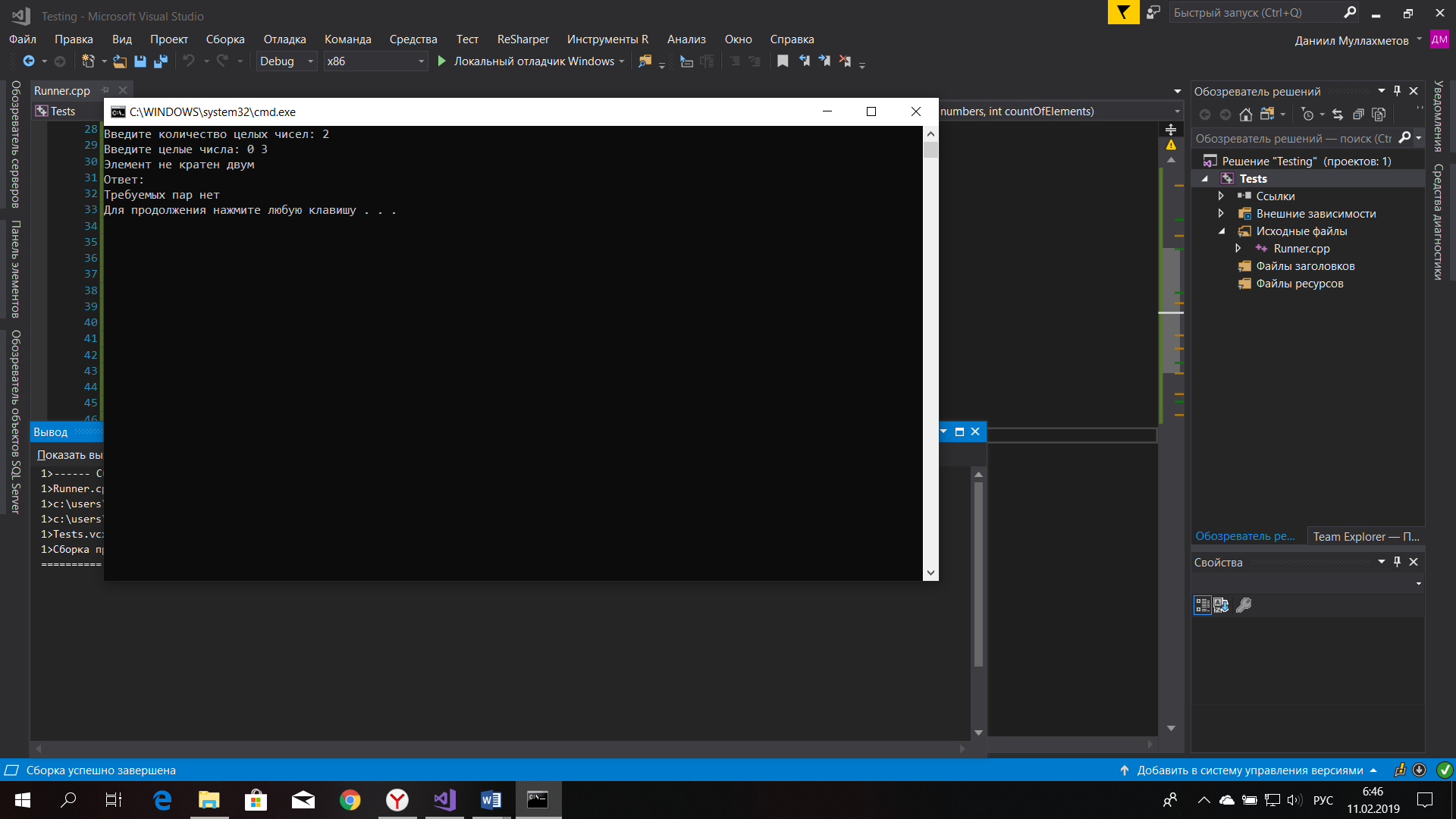


Рис.13

Шестой тестовый вариант

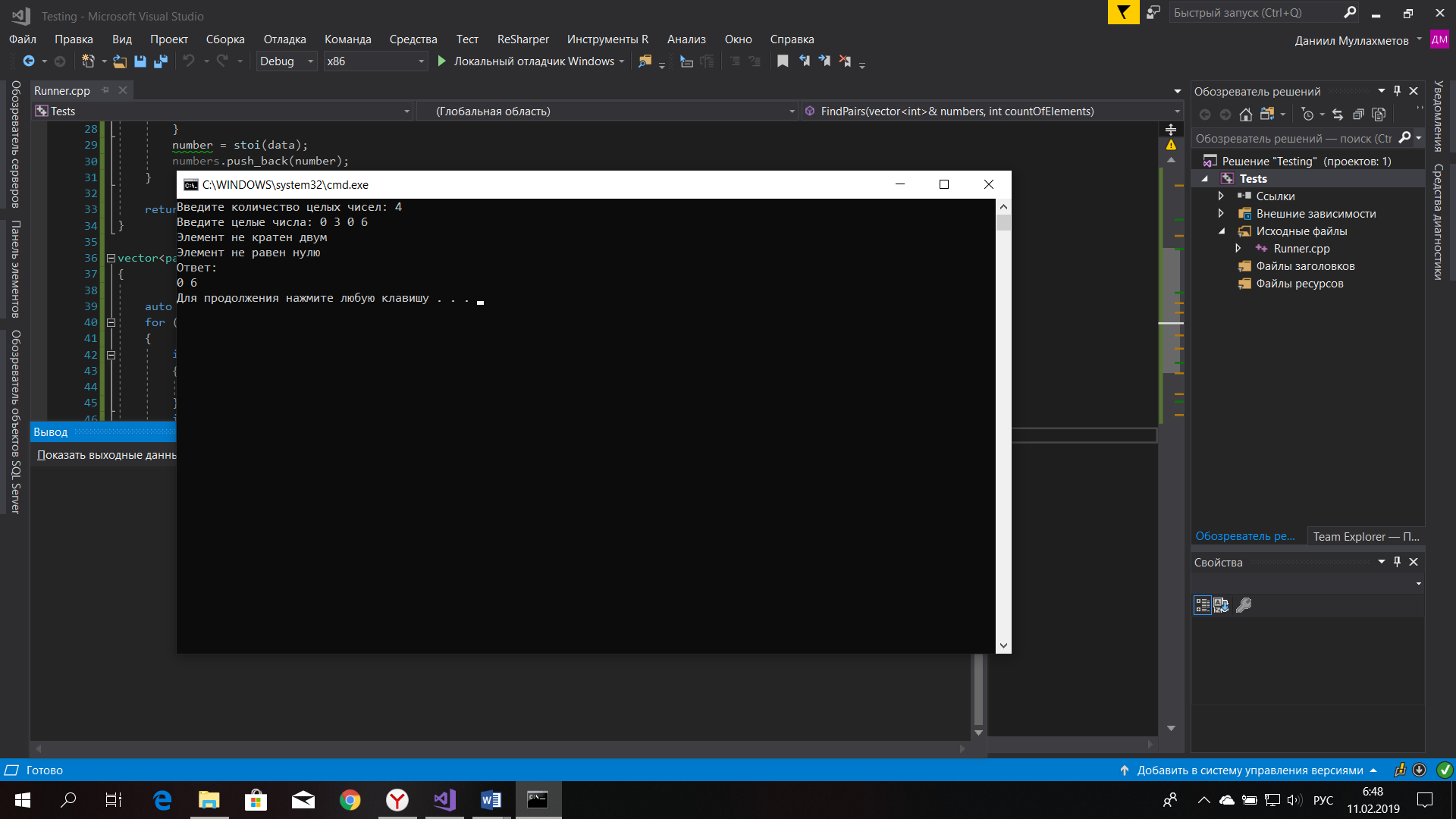


Рис.14

Седьмой тестовый вариант

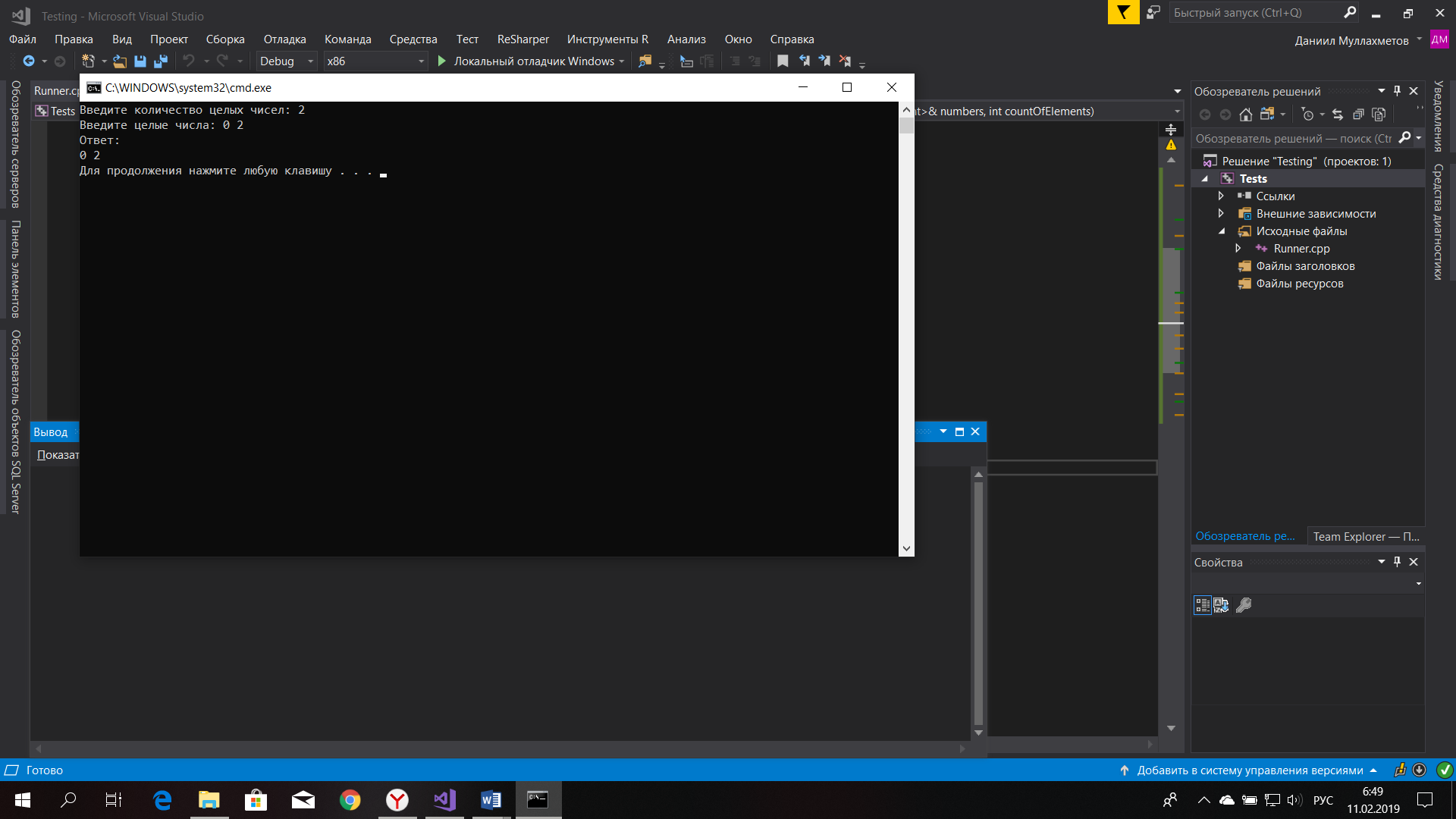


Рис.15

7. ТЕКСТ ПРОГРАММЫ

#include <iostream>

#include <vector>

#include <string>

using namespace std;

bool IsIntegerNumber(string number)

{

for (auto i = 0; i < number.size(); ++i)

{

if ((number[i] < '0' || number[i] > '9') && !(i == 0 && number[i] == '-'))

return false;

}

return true;

}

bool ReadData(vector<int> &numbers, int countOfElements)

{

string data;

int number;

for (auto i = 0; i < countOfElements; ++i)

{

cin >> data;

if (!IsIntegerNumber(data))

{

cout << "Введенный параметр не является целым числом" << endl;

return false;

}

number = stoi(data);

numbers.push\_back(number);

}

return true;

}

vector<pair<int, int>> FindPairs(vector<int> &numbers, int countOfElements)

{

auto pairs = vector<pair<int, int>>();

for (auto i = 0; i < countOfElements - 1; ++i)

{

if (numbers[i] == 0 && numbers[i + 1] % 2 == 0)

{

pairs.push\_back(pair<int, int>(numbers[i], numbers[i + 1]));

}

if (numbers[i] != 0) cout << "Элемент не равен нулю" << endl;

if (numbers[i + 1] % 2 != 0) cout << "Элемент не кратен двум" << endl;

}

return pairs;

}

void main()

{

setlocale(LC\_ALL, "Russian");

string data;

cout << "Введите количество целых чисел: ";

cin >> data;

if (!IsIntegerNumber(data))

{

cout << "Введенный параметр(кол-во элементов) не является целым числом" << endl;

return;

}

auto countOfElements = stoi(data);

auto numbers = vector<int>();

cout << "Введите целые числа: ";

if (!ReadData(numbers, countOfElements))

return;

auto pairs = FindPairs(numbers, countOfElements);

cout << "Ответ: " << endl;

if (!pairs.empty())

{

for (auto pair : pairs) {

cout << pair.first << " " << pair.second << endl;

}

}

else cout << "Требуемых пар нет" << endl;

}