Пензенский государственный университет

Факультет вычислительной техники

Кафедра САПР

### Тестирование собственного кода

Отчет по лабораторной работе № 1

по дисциплине

«Качество и тестирование программного обеспечения»

Выполнили: ст. гр. 11ВА1

Доброжанская П.С.

Проверила:

доцент каф. САПР Эпп В.В.

Пенза, 2014

1. Цель работы – тестирование собственного кода.
2. Задание:

Дан массив размера *N* (*N* — четное число). Поменять местами его первый элемент со вторым, третий — с четвертым и т. д., проверяя при этом элемент стоящий на четном месте, и если он больше 0, то прибавлять к нему число К.

## Спецификация

Спецификация

1. Общее описание

Программа состоит из одного модуля – «Form1», в котором помимо взаимодействия с пользователем так же реализован весь функционал программы. После ввода или загрузки входных данных происходит обработка полученных данных с последующей возможностью вывода результата на экран и в файл.

1. Описание интерфейса
   1. Входные данные
      1. Набор целых чисел, как положительных, так и отрицательных.
      2. Входные данные могут быть введены в поле ввода при помощи клавиатуры, или же прочтены из файла с именем out.txt при нажатии кнопки «Ввод из файла», при условии, что он существует, в противном случае будет выведено сообщение ERROR03.
   2. Выходные данные
      1. Вывод в файл.

Файл с именем output.txt. Данный файл служит для хранения результата работы программы – набора символов, включающего в себя целые числа, как положительные, так и отрицательные. Название файла зашито в программном коде и не может быть изменено. Результат выводится автоматически, и для ввода с клавиатуры и для ввода из файла. При условии, если файл существует, набор будет перезаписан. В противном случае будет создан файл с именем output.txt, в который будет записан результат. Данный файл хранится в той же папке, что и сама программа.

* + 1. Вывод на экран.

Набор символов, включающий в себя целые числа, как положительные, так и отрицательные. Данный набор может быть выведен в поле вывода при условии, что входные данные загружены либо введены вручную и нажата кнопка «Выполнить».

* + 1. Сообщения об ошибках, выдаваемые программой.

ERROR01 - Длина массива не соответствует введенному.

ERROR02 - Введите размерность и произвольное число.

ERROR03 — Ошибка с открытием или расположением файла.

ERROR04— Проверьте исходные данные. Массив целочисленных элементов.

ERROR 05 - Введите произвольное число.

ERROR 06 — Введите четное число.

ERROR07 - Длина массива не соответствует введенному! Дополните массив до нужного размера.

ERROR 08 - Длина массива не соответствует введенному! Вы ввели большое количество символов

ERROR09 – Выход за пределы границ типа.

* 1. Описание файлов входящих в пакет программы

out.txt – файл содержащий входные данные.

output.txt – файл для записи результата.

ktpo1.exe – основная программа.

* 1. Интерфейс пользователя

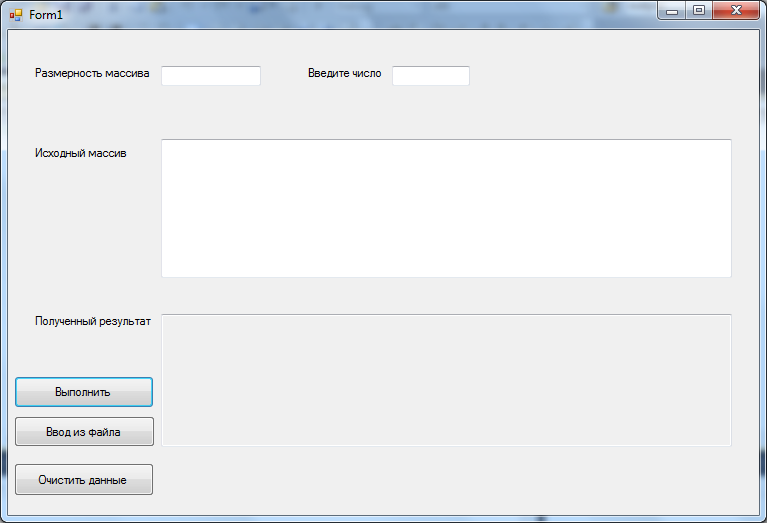


Рис.1 Окно программы

Кнопки «Выполнить», «Ввод из файла», позволяют получить результат, выведенный в текстовое поле «Полученный результат». Для корректной работы программы, данные вводятся в поле «Исходный массив» через пробел, значения, которых могут принимать как отрицательные, 0, положительные числа. При вводе произвольного числа, никаких ограничений не накладывается, кроме ситуации, когда поле ввода не заполнено. При заполнении поля «Размерность массива», накладывается условие, что число должно быть четным.

Кнопка «Очистить данные» позволяет очистить поля ввода, для последующего применения.

Автоматически данные выводятся и в текстовый файл.

1. Описание архитектуры

Программа состоит из одного модуля. Модуль содержит 2 функции и 3 обработчика события.

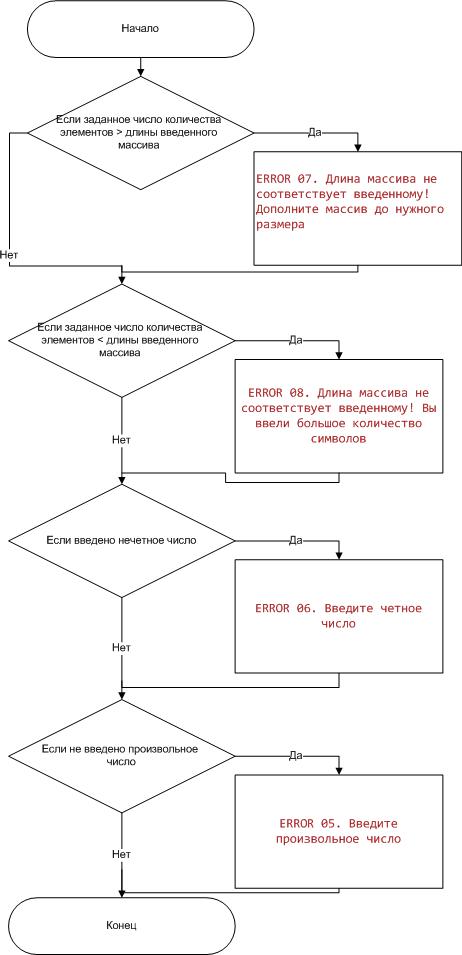


Рис. 2 Алгоритм проверок ввода данных

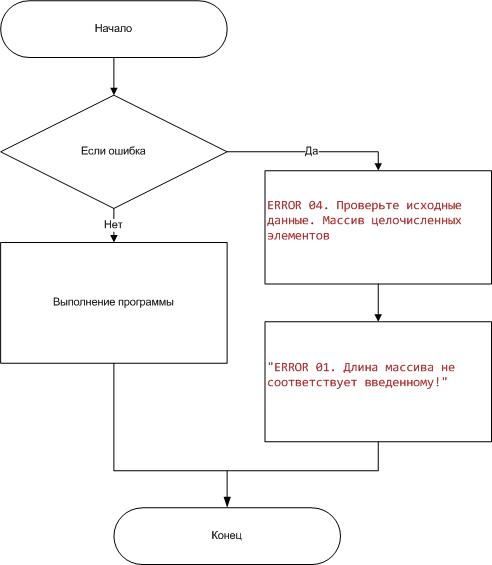


Рис. 3 Обработка исключительной ситуации в выполнении программы

1. Функциональные требования
   1. Требования к программе

Программа выполняет следующие действия:

* + 1. Функция обмена.

Меняет местами элементы массива. Первый элемент со вторым, третий — с четвертым и т. д., проверяя при этом элемент, стоящий на четном месте, и если он больше 0, то прибавлять к нему число К.

* + 1. Функция ручного ввода входных данных.

Позволяет пользователю вводить входные данные в поле ввода.

* + 1. Ввод числа К.

Число может быть как положительное, так и отрицательное, либо 0. Никаких ограничений не накладывается.

* + 1. Функция чтения входных данных из файла.

Позволяет пользователю читать входные данные из файла out.txt.

* + 1. Функция вывода результата на экран.

Позволяет пользователю вывести результат работы программы в поле вывода.

* 1. Требования к вводу с клавиатуры
     1. Каждое вводимое число должно быть отделено пробелом и не содержать некорректных символов. Некорректными считаются символы отличные от чисел от 0 до 9 и знака `-`, причем символ `-` должен стоять перед числом и без наличия пробелов между ними, например -5. Если же пользователь вводит один из некорректных символов, программа выведет ERROR04— Проверьте исходные данные. Массив целочисленных элементов.
     2. Ввод числа К. Ограничений нет. Если число не введено в поле, то выведется сообщение ERROR05 – Введите произвольное число.
     3. Ввод размерности массива. Число должно быть четным, если число введено нечетное выводится сообщение ERROR 06 — Введите четное число.
  2. Требования к вводу из файла

Для чтения массива из файла, необходимо, что бы файл out.txt был размещен в папке с программой, если же файл отсутствует, то будет выведена ERROR03 — Ошибка с открытием или расположением файла

Тесты

|  |  |  |
| --- | --- | --- |
|  | 1 | Приоритет |
| Идея:  INFO: | Проверка пункта 2.1.1. Ввод положительных и отрицательных чисел.  -1 2 0 -8 8 7 | |
| RevisionHistory | | |
| Создан 16.04.14 | | |
| Процедура | Ожидаемый результат | |
| 1. Запуск ktpo1.exe 2. Ввод чисел в поле ввода | Вывод:  4 -1 -8 0 7 10 | |
| Полученный результат: 4 -1 -8 0 7 10 | | |

|  |  |  |
| --- | --- | --- |
|  | 2 | Приоритет |
| Идея:  INFO:  out.txt отсутствует | Проверка пункта 2.1.3.  Ввод из файла при условии, что файл отсутствует. | |
| RevisionHistory | | |
| Создан 16.04.14 | | |
| Процедура | Ожидаемый результат | |
| 1. Запуск ktpo1.exe 2. Нажатие «Открыть файл» | Вывод: ERROR03 — Ошибка с открытием или расположением файла. | |
| Полученный результат: ERROR03 — Ошибка с открытием или расположением файла. | | |

|  |  |  |
| --- | --- | --- |
|  | 3 | Приоритет |
| Идея:  INFO: | Проверка пункта 2.2.1. Вывод результата в файл.  -1 2 0 -8 8 7 | |
| RevisionHistory | | |
| Создан 16.04.14 | | |
| Процедура | Ожидаемый результат | |
| 1. Запуск ktpo1.exe 2. Нажатие «Открыть файл» 3. Нажать «Вывод в файл» | файл output.txt содержит 4 -1 -8 0 7 10 | |
| Полученный результат: файл output.txt содержит 4 -1 -8 0 7 10 | | |

|  |  |  |
| --- | --- | --- |
|  | 3 | Приоритет |
| Идея:  INFO:  output.txt отсутствует | Проверка пункта 2.2.1. Вывод результата отсутствующий файл  . -1 2 0 -8 8 7 | |
| RevisionHistory | | |
| Создан 16.04.14 | | |
| Процедура | Ожидаемый результат | |
| 1. Запуск ktpo1.exe 2. Нажатие «Открыть файл» 3. Нажать «Вывод в файл» | файл output.txtсоздан и содержит 4 -1 -8 0 7 10 | |
| Полученный результат: файл output.txt создан и содержит 4 -1 -8 0 7 10 | | |

|  |  |  |
| --- | --- | --- |
|  | 4 | Приоритет |
| Идея:  INFO: | Проверка пункта 4.2.2. Вывод недопустимых символов с клавиатуры.  -1 о 3 4 5 0 | |
| RevisionHistory | | |
| Создан 16.04.14 | | |
| Процедура | Ожидаемый результат | |
| 1. Запуск ktpo1.exe 2. Ввод данных в поле ввода | ERROR04— Проверьте исходные данные. Массив целочисленных элементов. | |
| Полученный результат: ERROR04— Проверьте исходные данные. Массив целочисленных элементов. | | |

Текст программы

Приложение А

(обязательное)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.IO;

using System.Windows.Forms;

namespace ktpo1

{

public partial class Form1 : Form

{

FileStream fout = new FileStream("file.txt", FileMode.Open);

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

int temp;

try

{

textBox3.Clear();

int c = int.Parse(textBox1.Text); // Парсим значение из текстового поля textBox1.

int k = int.Parse(textBox4.Text);

string[] sNums = textBox2.Text.Split(' '); // Разбиваем текст из текстового поля textBox2, на массив строк, разделителем является пробел.

if (c > sNums.Length)

{

MessageBox.Show("ERROR 07. Длина массива не соответствует введенному! Дополните массив до нужного размера");

return;

}

if (c < sNums.Length)

{

MessageBox.Show("ERROR 08. Длина массива не соответствует введенному! Вы ввели большое количество символов");

return;

}

if (c % 2 != 0)

{

MessageBox.Show("ERROR 06. Введите четное число");

textBox1.Clear();

return;

}

if (textBox4.Text == null)

{

MessageBox.Show("ERROR 05. Введите произвольное число");

}

int[] nums = new int[c]; // объявляем новый массив с количеством элементов = c.

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

nums[i] = int.Parse(sNums[i]);

for (int i = 1; i <= c; i = i + 2)

{

if (nums[i] > 0)

{

nums[i] = nums[i] + k;

}

}

for (int i = 0; i < c / 2; i++)

{

temp = nums[i \* 2 + 1];

nums[i \* 2 + 1] = nums[i \* 2];

nums[i \* 2] = temp;

}

for (int i = 0; i < nums.Length; i++)

{

textBox3.Text += nums[i].ToString() + " ";

}

FileStream fs = new FileStream("output.txt", FileMode.OpenOrCreate, FileAccess.Write);

StreamWriter sw = new StreamWriter(fs);

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

{

sw.WriteLine(nums[i] + " ");

}

sw.Close();

}

catch (IndexOutOfRangeException)

{

MessageBox.Show("ERROR 01. Длина массива не соответствует введенному!");

return;

}

catch

{

MessageBox.Show("ERROR 04. Проверьте исходные данные. Массив целочисленных элементов");

}

}

//ВВОД ИЗ ФАЙЛА

private void button3\_Click(object sender, EventArgs e)

{

try

{

textBox3.Clear();

int c = int.Parse(textBox1.Text);

int k = Convert.ToInt32(textBox4.Text);

textBox2.Text = File.ReadAllText("out.txt");

string[] sNums = textBox2.Text.Split(' '); // Разбиваем текст из текстового поля textBox2, на массив строк, разделителем является пробел.

if (c > sNums.Length)

{

MessageBox.Show("ERROR 01. Длина массива не соответствует введенному!");

return;

}

//if (c < sNums.Length)

//{

// MessageBox.Show("Длина массива не соответствует введенному!");

// return;

//}

int[] nums = new int[c]; // объявляем новый массив с количеством элементов = c.

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

nums[i] = int.Parse(sNums[i]);

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

{

if ((nums[i] % 2 == 0) && (nums[i] > 0))

{

nums[i] = nums[i] + k;

}

}

for (int i = 0; i < c / 2; i++)

{

int temp = nums[i \* 2 + 1];

nums[i \* 2 + 1] = nums[i \* 2];

nums[i \* 2] = temp;

}

for (int i = 0; i < nums.Length; i++)

{

textBox3.Text += nums[i].ToString() + " ";

}

}

catch (IOException ex)

{

//MessageBox.Show("Ошибка открытия файла");

MessageBox.Show("ERROR 03. ",ex.Message);

}

catch

{

MessageBox.Show("ERROR 02. Введите размерность и произвольное число");

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

}

}

}}

РЕЗУЛЬТАТЫИСПЫТАНИЙ

Приложение В

(обязательное)

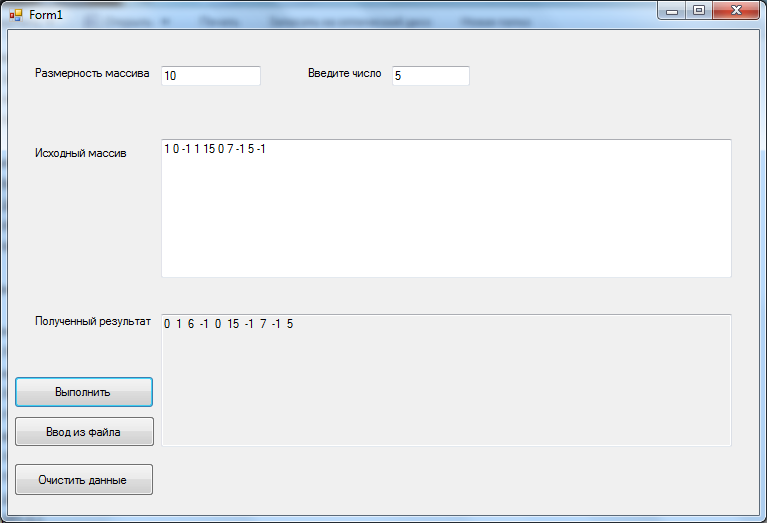


Рис. 4 Результат выполнения программы, ввод с клавиатуры

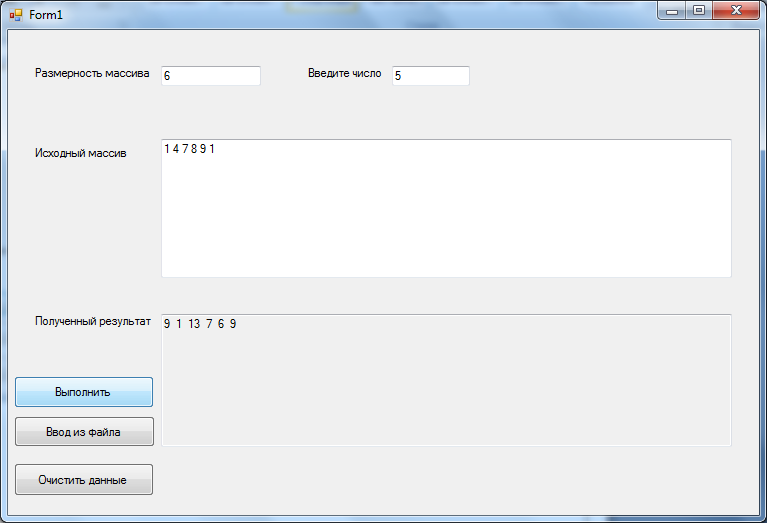


Рис. 5 Результат выполнения массива, только положительных чисел с клавиатуры

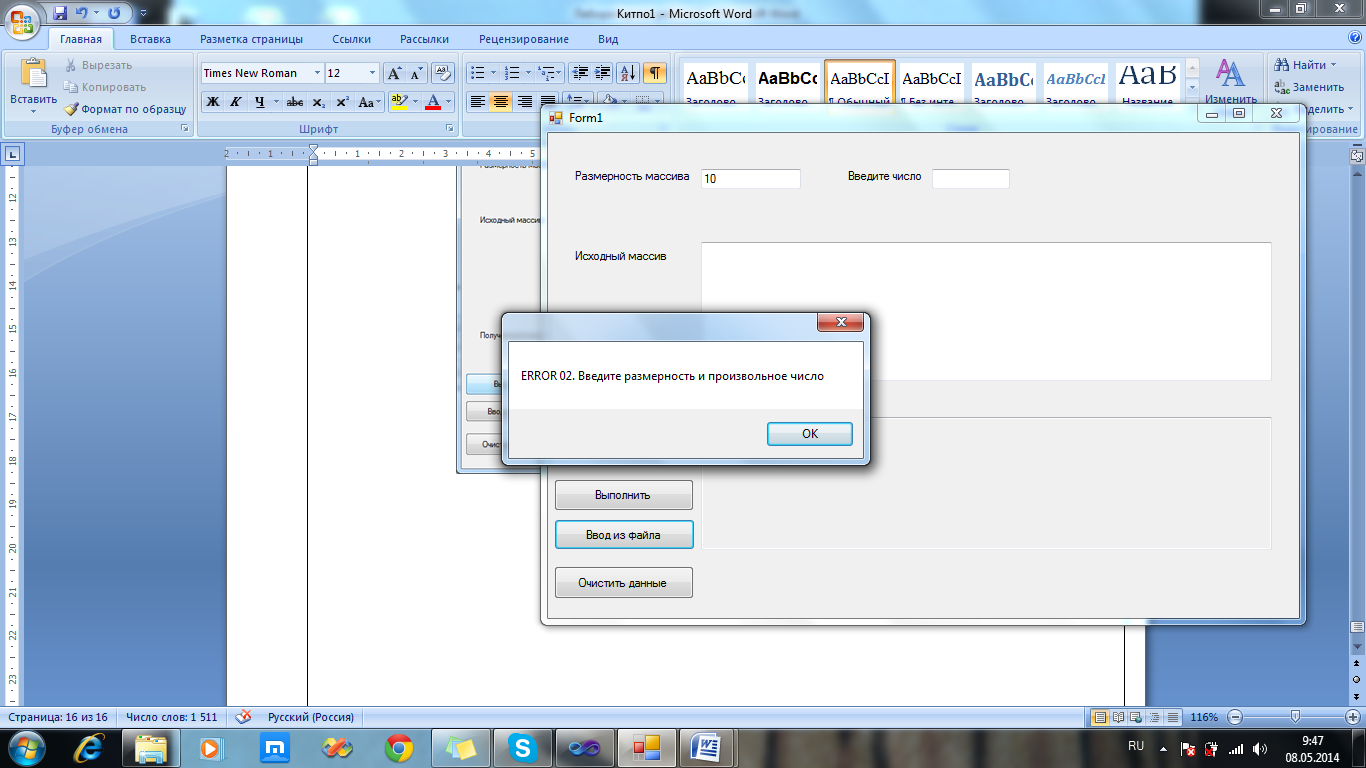


Рис. 6 Результат, если не все поля заполнены. Ввод из файла

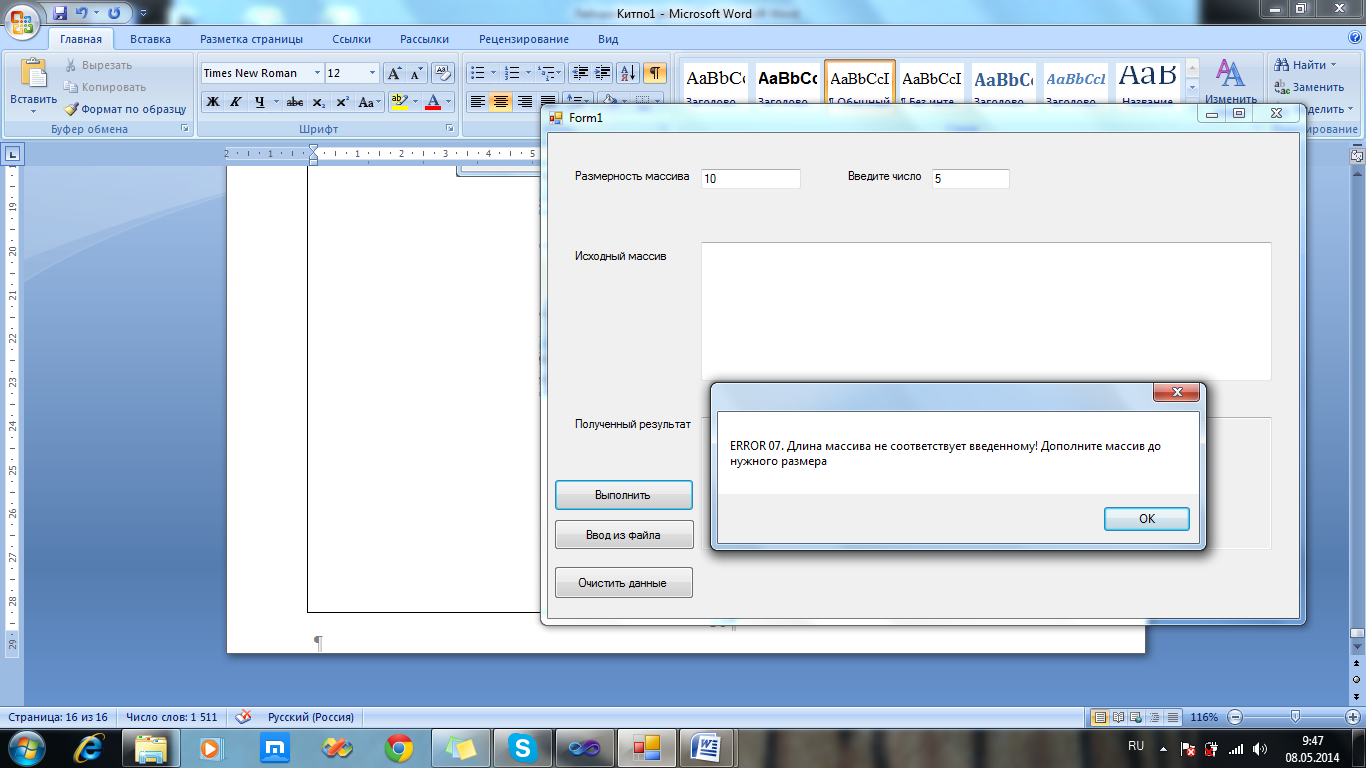


Рис. 7 Результат, если не введен массив в поле

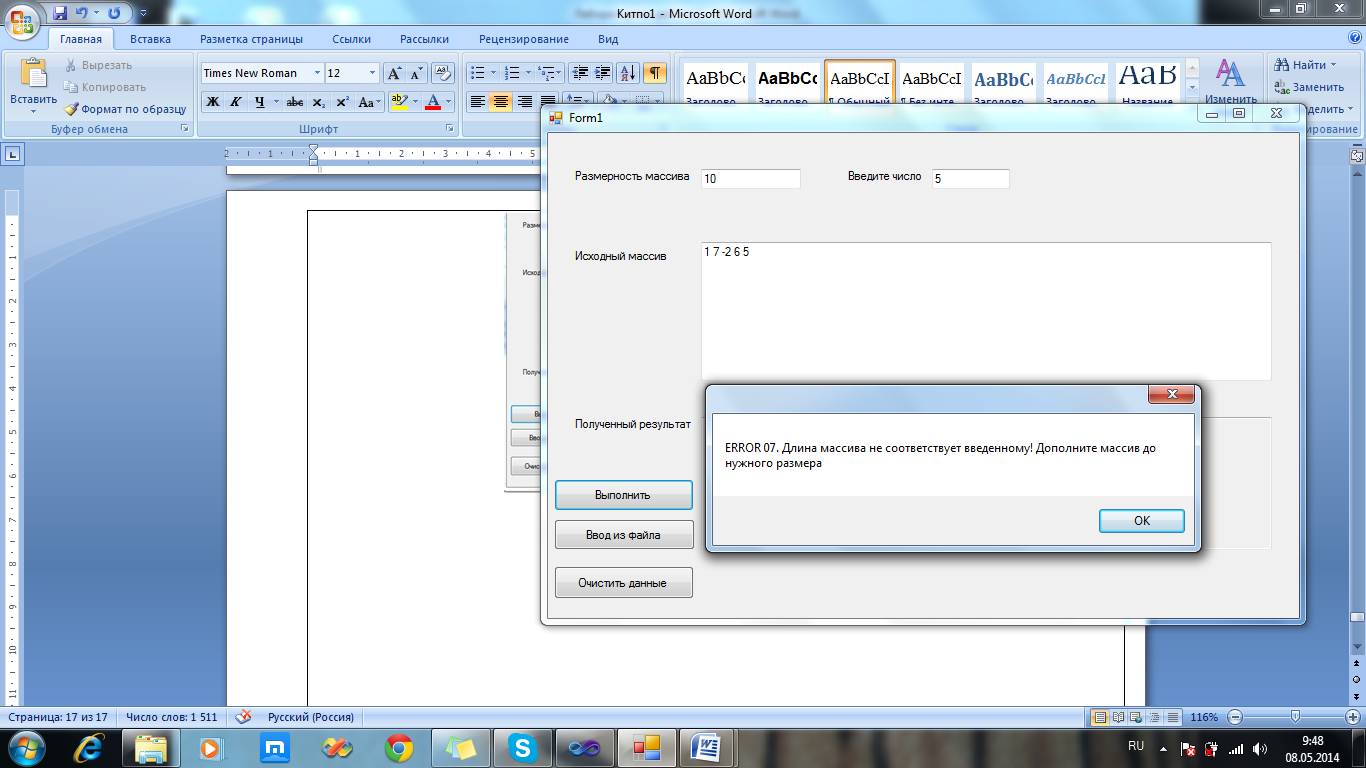


Рис. 8 Результат, если число введенных элементов массива не совпадает с его размерностью

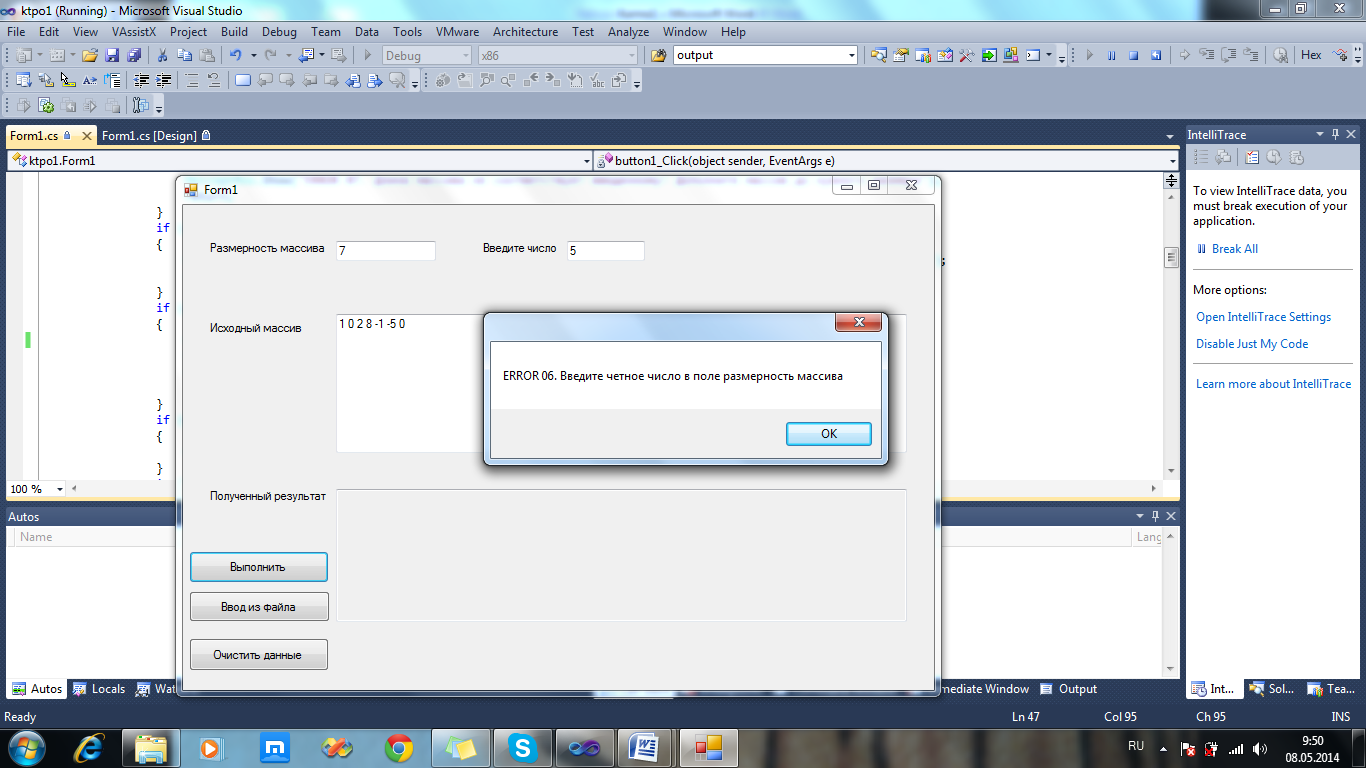


Рис. 9 Результат, если размерность массива не четное число