Тест-требования

1. Тестирования ввода размера массива:
   1. Ввод не числа.
   2. Ввод отрицательного числа.
   3. Ввод не целого числа.
2. Тестирование на заполнения элементов массива:
   1. Ввод пустой строки.
   2. Ввод элементов массива.
3. Тестирование сохранения данных в файл в каталоге программы:
   1. Сохранение данных без созданного файла.
   2. Сохранение данных с созданным файлом.

Тест-план

Тестовый пример 1.

Номер тест требования 1.1.

Описание теста:

При запросе размера массива указать не число.

Входные данные:

Размер массива «ыывыв»

Ожидаемый результат:

Программа обработает исключение ошибки ввода данных и запросит подтверждение повторения программы.

Сценарий:

* Запустить программу.
* При запросе от программы размера массива указать не число.

Тестовый пример 2.

Номер тест требования 1.2.

Описание теста:

При запросе размера массива указать отрицательное число.

Входные данные:

Размер массива «-2»

Ожидаемый результат:

Программа обработает исключение ошибки переполнения и запросит подтверждение повторения программы.

Сценарий

* Запустить программу.
* При запросе от программы размера массива указать отрицательное число.

Тестовый пример 3.

Номер тест требования 1.3.

Описание теста:

При запросе размера массива указать не целое число.

Входные данные:

Размер массива «2.2»

Ожидаемый результат:

Программа обработает исключение ошибки неверного формата данных и запросит подтверждение повторения программы.

Сценарий:

* Запустить программу.
* При запросе размера массива указать не целое число.

Тестовый пример 4.

Номер тест требования 2.1.

Описание теста:

При запросе на ввод элемента массива указать пустую строку.

Входные данные:

Размер массива «3»; «Утин»; Имя «Дмитрий»; Возраст «19»; Фамилия «Кенкебашвили»; Имя «Илья»; Возраст «22»; Фамилия « »; Имя « »; Возраст « »

Ожидаемый результат:

Программа продолжит работать дальше и оставит поля пустыми.

Сценарий:

* Запустить программу.
* Ввести размер массива.
* Заполнить один элемент массива пустыми строками
* Заполнить оставшиеся элементы массива.
* Завершить программу.

Тестовый пример 5.

Номер тест требования 2.2.

Описание теста:

При запросе на ввод элементов заполнить все запрашиваемые элементы массива.

Входные данные:

Размер массива «3»; Фамилия «Утин»; Имя «Дмитрий»; Возраст «19»; Фамилия «Кенкебашвили»; Имя «Илья»; Возраст «22»; Фамилия «Тимофеев»; Имя «Никита»; Возраст «20»

Ожидаемый результат:

Программа проведет сортировку элементов массива по Фамилии и Имя в алфавитном порядке, отобразит отсортированный массив и сохранит его в файл в корне программы.

Сценарий:

* Запустить программу.
* Ввести размер массива.
* Заполнить элементы массива.
* Проверить выведенный отсортированный массив.
* Завершить программу.
* Проверить сохранённый файл с отсортированный массивом.

Тестовый пример 6.

Номер тест требования 3.1.

Описание теста:

Проверить что при отсутствии файл «Persons» в корневой папке программы, она создает новый файл и записывает в него результат.

Входные данные:

Размер массива «3»; Фамилия «Утин»; Имя «Дмитрий»; Возраст «19»; Фамилия «Кенкебашвили»; Имя «Илья»; Возраст «22»; Фамилия «Тимофеев»; Имя «Никита»; Возраст «20»

Ожидаемый результат:

Программа создаст и запишет результат в новый файл «Persons».

Сценарий:

* Запустить программу.
* Ввести размер массива.
* Заполнить элементы массива.
* Проверить выведенный отсортированный массив.
* Завершить программу.
* Проверить сохранённый файл с отсортированный массивом.

Тестовый пример 7.

Номер тест требования 3.2.

Описание теста:

Проверить что программа при наличии файла перезаписывает его.

Входные данные:

Размер массива «3»; Фамилия «Утин»; Имя «Дмитрий»; Возраст «19»; Фамилия «Кенкебашвили»; Имя «Илья»; Возраст «22»; Фамилия «Тимофеев»; Имя «Никита»; Возраст «20»

Ожидаемый результат:

Программа перезапишет информацию в уже созданном файле.

Сценарий:

* Запустить программу.
* Ввести размер массива.
* Заполнить элементы массива.
* Проверить выведенный отсортированный массив.
* Завершить программу.
* Проверить сохранённый файл с отсортированный массивом.

Тест 1.

Входные данные:

Указанно не число при запросе размера массива.

Выходные данные:

Обработанное исключение и запрос на продолжение представлено на рисунке 1.

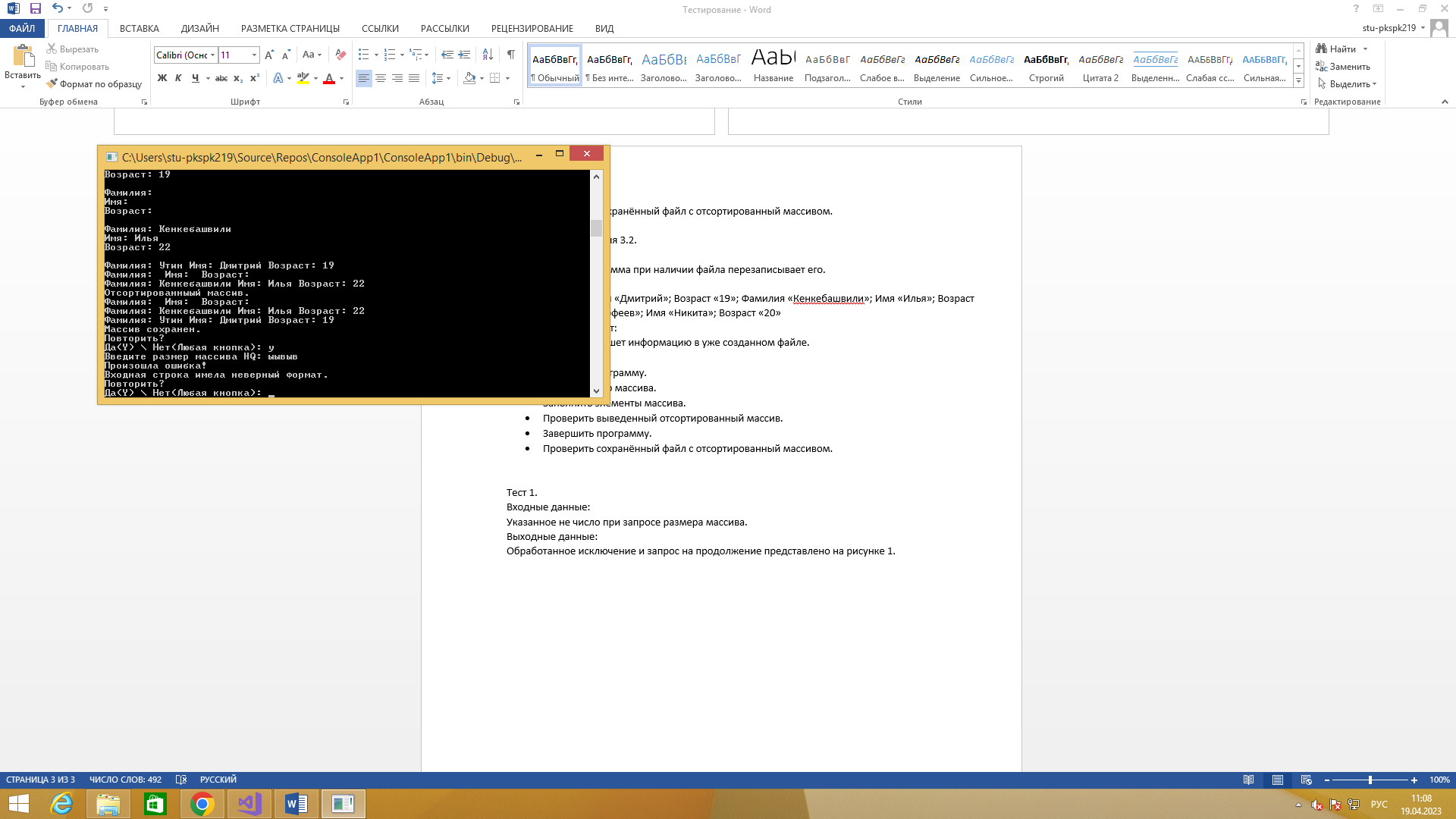


Рисунок 1 – Результат тестирования 1 тестового примера

Тест 2.

Входные данные:

Указано отрицательное число при запросе размера массива.

Выходные данные:

Обработанное исключение и запрос на продолжение представлено на рисунке 2.

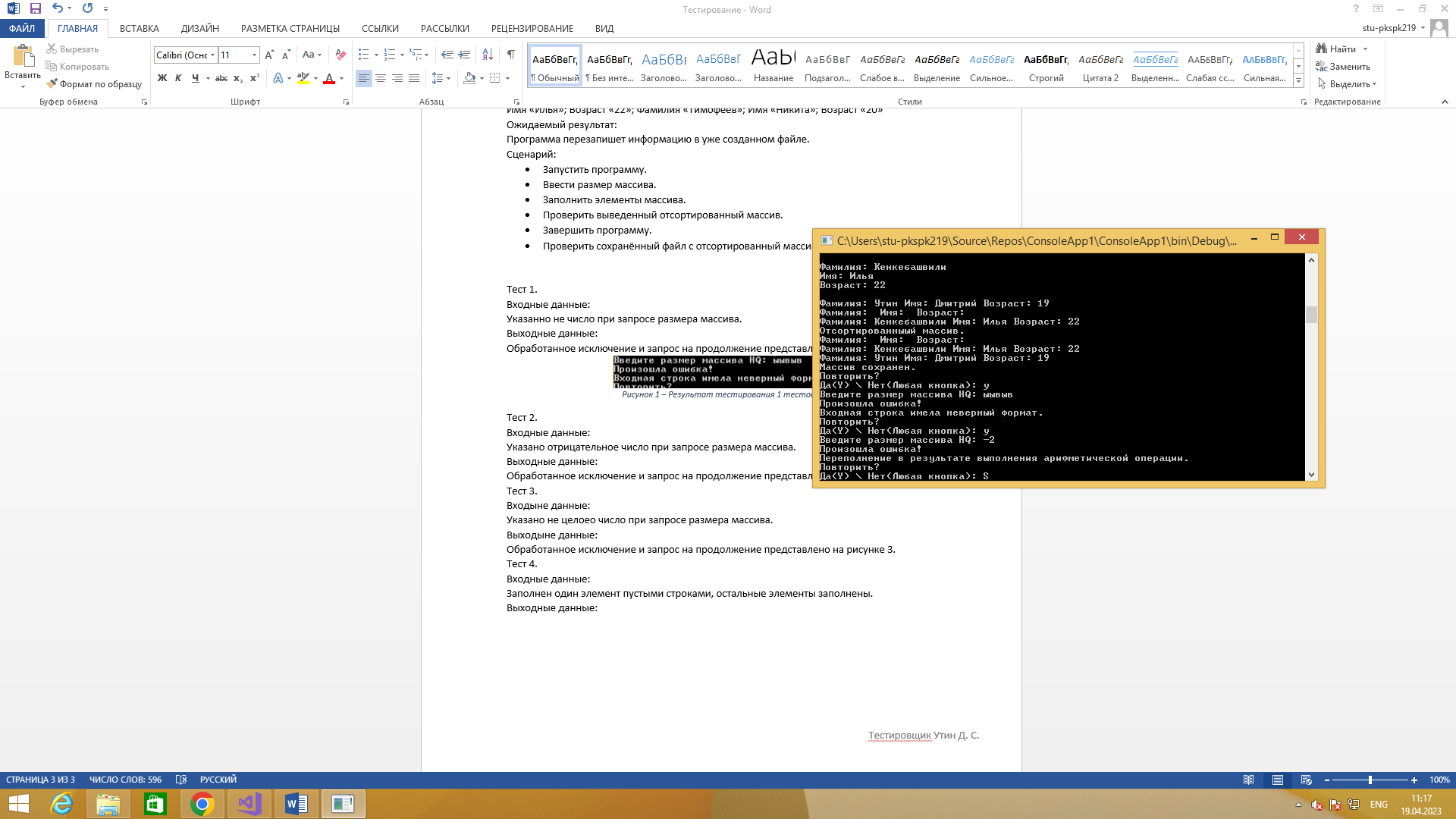


Рисунок 2 – Результат тестирования 2 тестового примера

Тест 3.

Входыне данные:

Указано не целоео число при запросе размера массива.

Выходыне данные:

Обработанное исключение и запрос на продолжение представлено на рисунке 3.

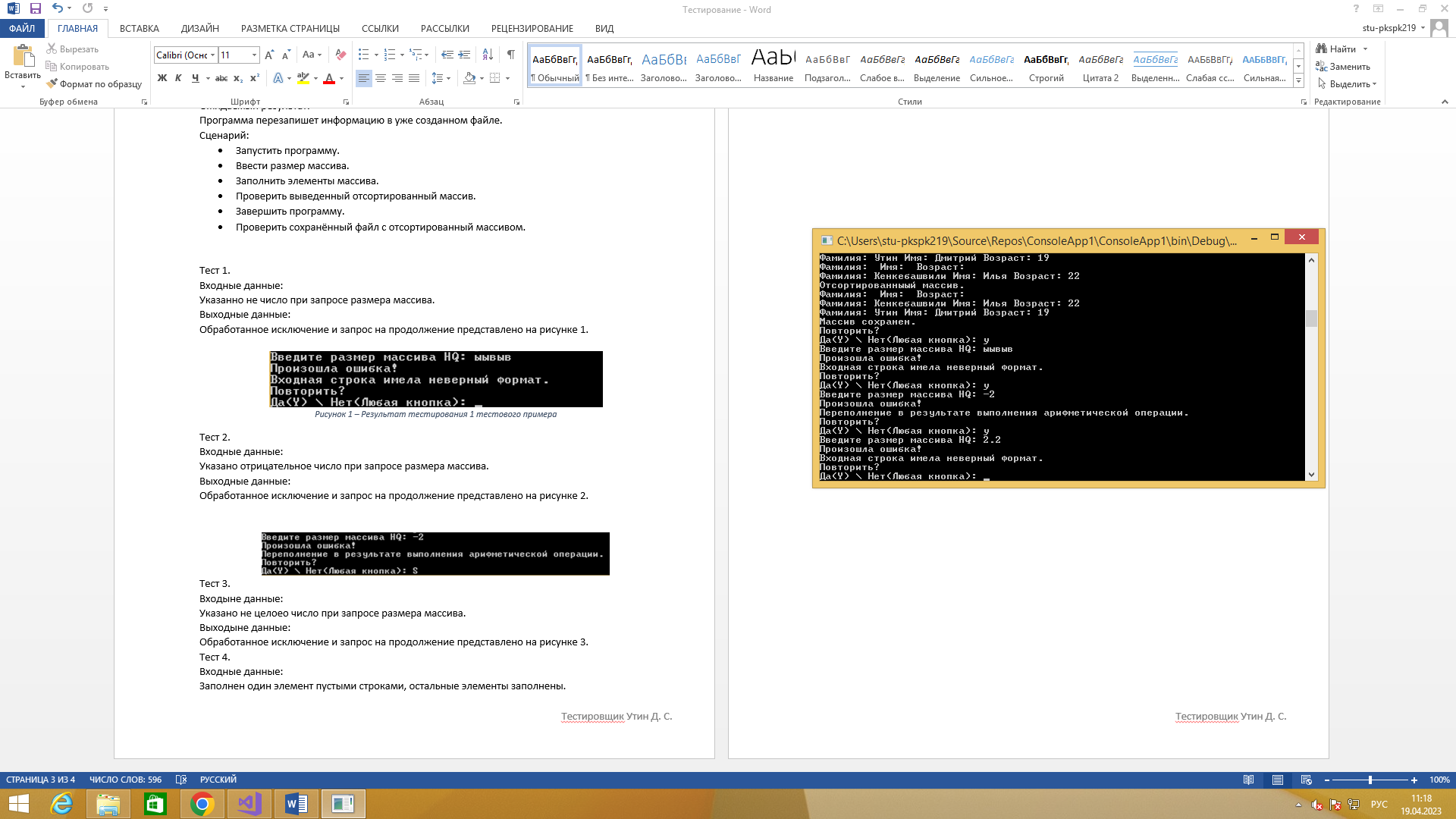


Рисунок 3 – Результат тестирования 3 тестового примера

Тест 4.

Входные данные:

Заполнен один элемент пустыми строками, остальные элементы заполнены в массиве размером 3

Выходные данные:

Программа продолжила работу, отсортировала массив, представленный на рисунке 4 и сохранила его в файл.

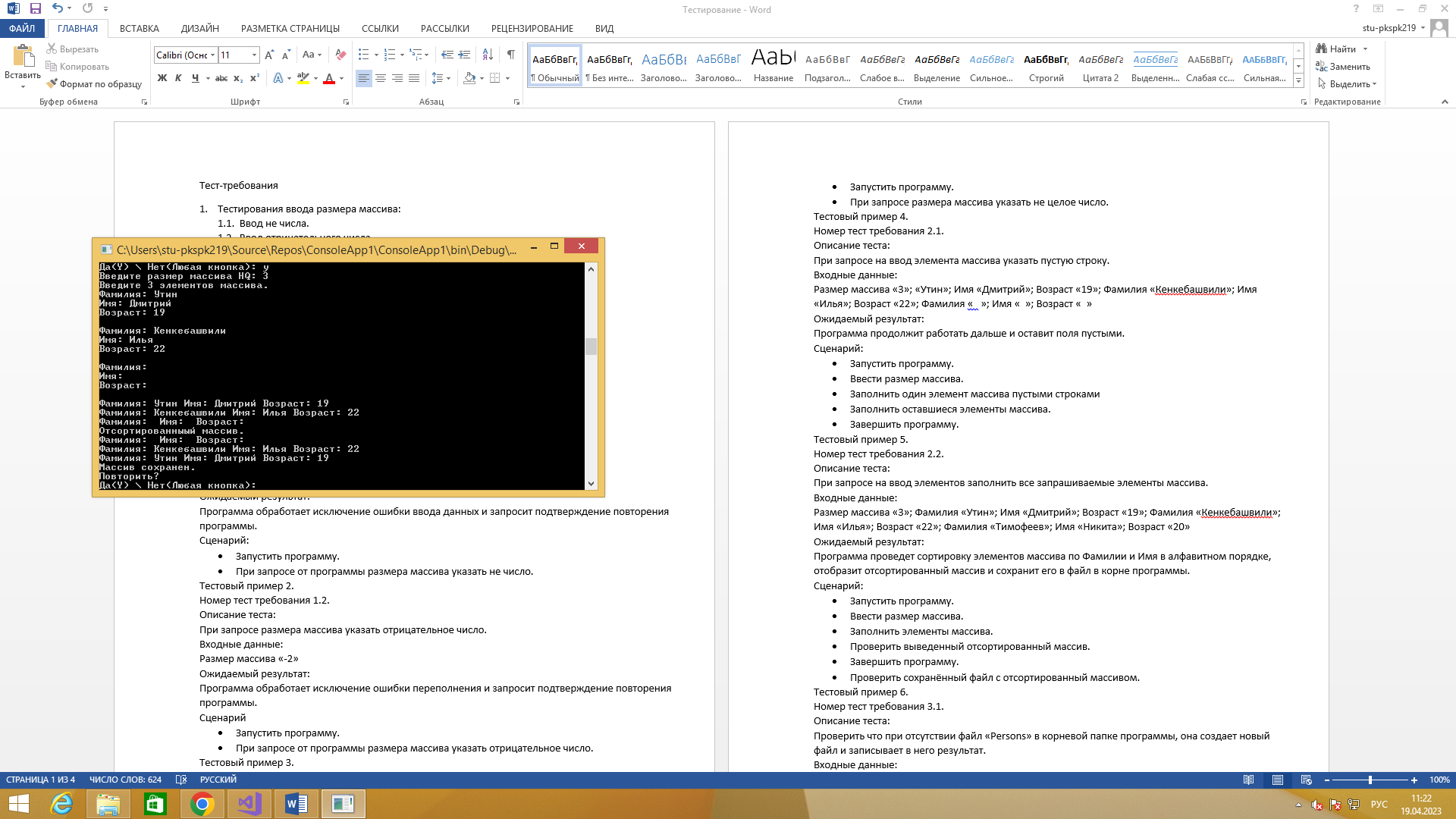


Рисунок 4 – Результат тестирования 4 тестового примера

Тест 5.

Входные данные:

Заполненные все элементы массива размером 3.

Выходные данные:

Программа вывела введённый массив, отсортировала его, снова вывела результат, представленный на рисунке 5 и записала его в файл.

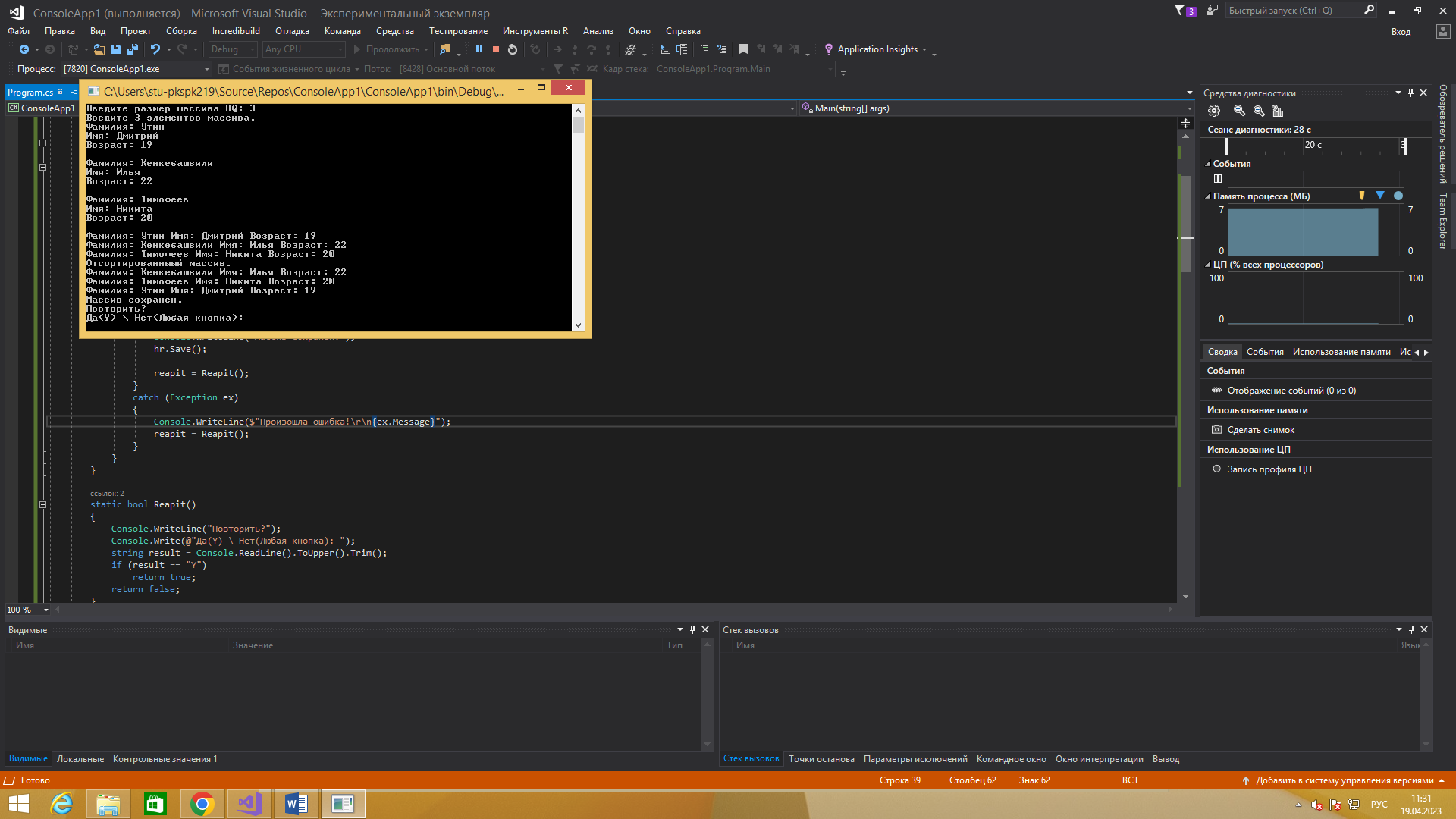


Рисунок 5 – Результат тестирования 5 тестового примера

Тест 6.

Входные данные:

Удаление файла «Persons» в корневой папке программы. И ее последующее использование.

Выходные данные:

Программа отсортирует массив размером 3, создаст файл и запишет в него результат, созданный файл представлен на рисунке 6.

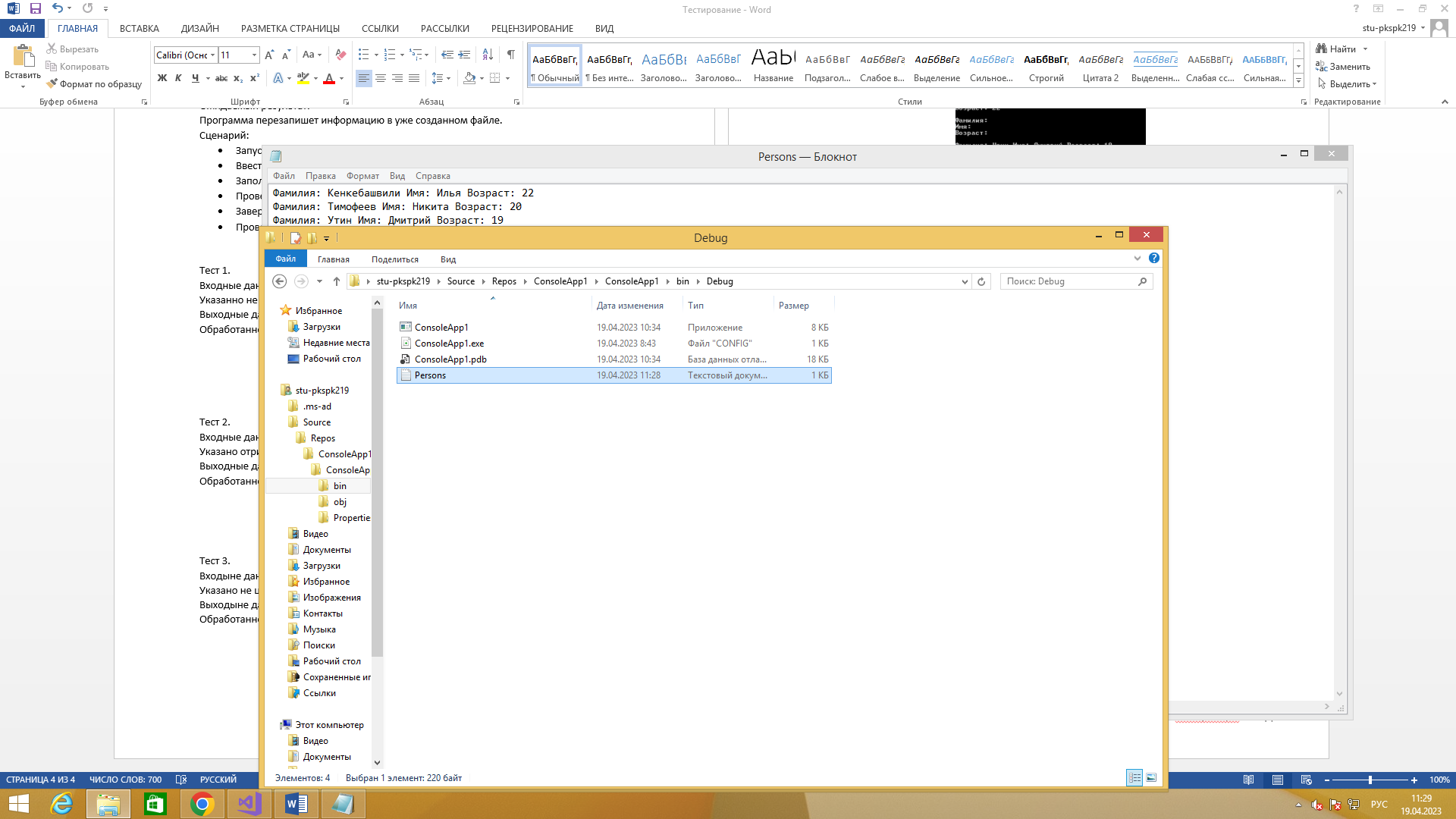


Рисунок 6 – Результат тестирования 6 тестового примера

Тест 7.

Входные данные:

Создать, если файл «Persons» отсевает, файл в корневой папке программы.

Выходные данные:

Программа перезаписала файл, представленный на рисунке 7.

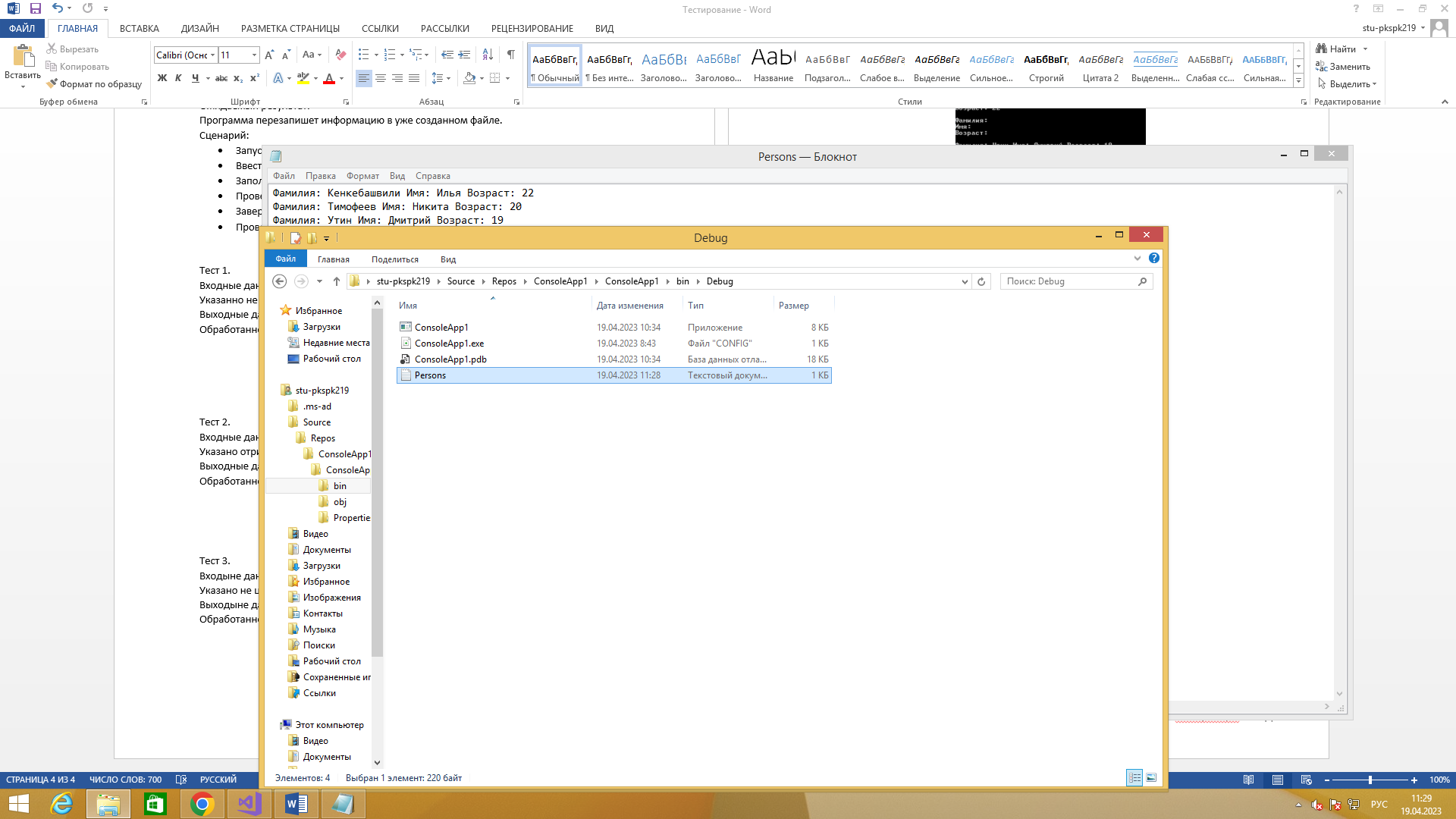


Рисунок 7 – Результат тестирования 7 тестового примера