**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

**Отчет**

**По лабораторной работе №7**

По дисциплине: Аппаратные основы интеллектуальных систем

На тему: Моделирование ассоциативного процессора с применением последовательных (рекуррентных) алгоритмов.

Выполнил:                                   Щур А. А.

Группа 821703

Проверил:                                    Захаров В.В.

**Минск 2019**

Цель работы: освоение навыков построения и проверки моделей ассоциативного процессора с применением рекуррентных алгоритмов.

**Задание**

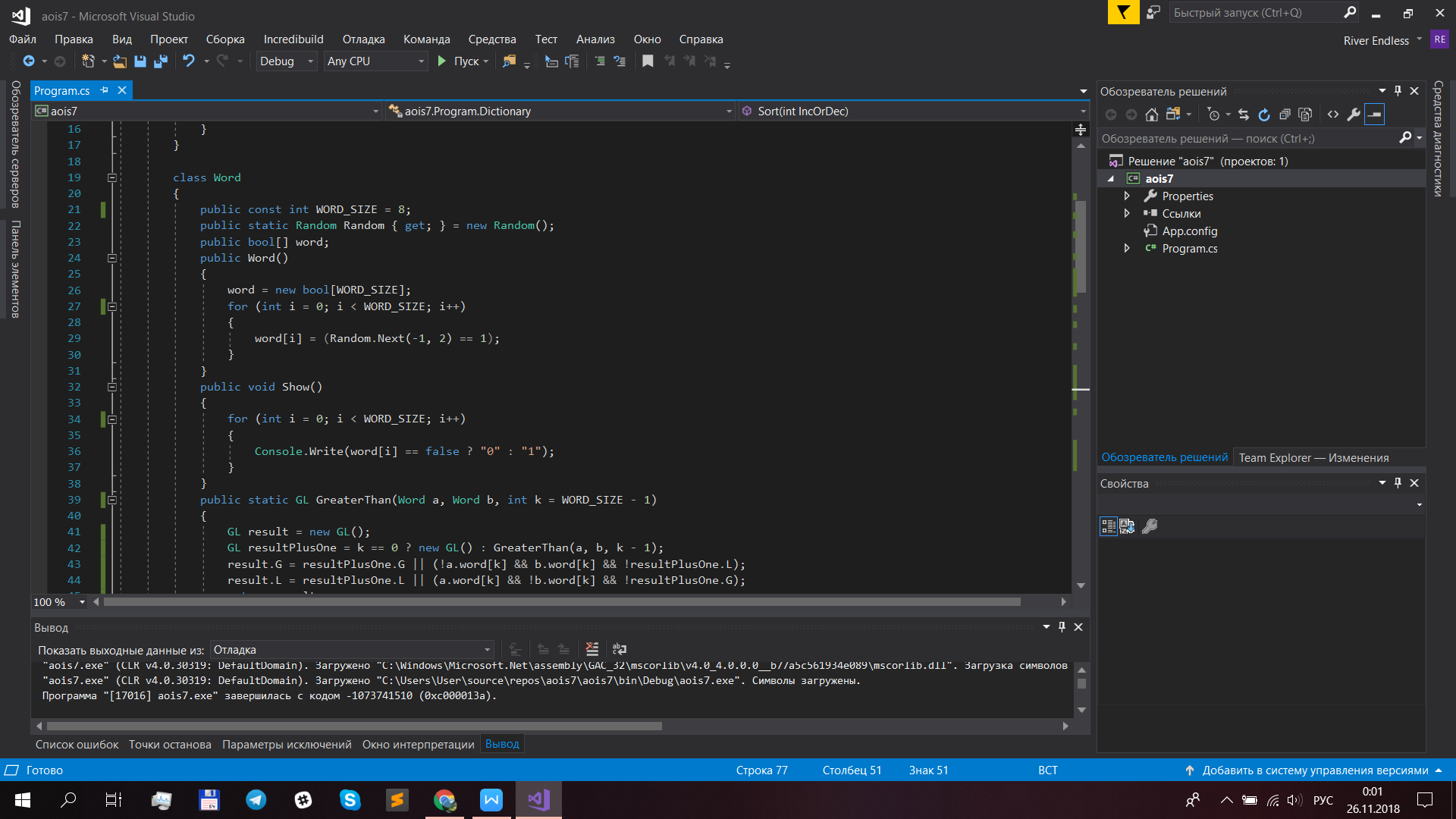
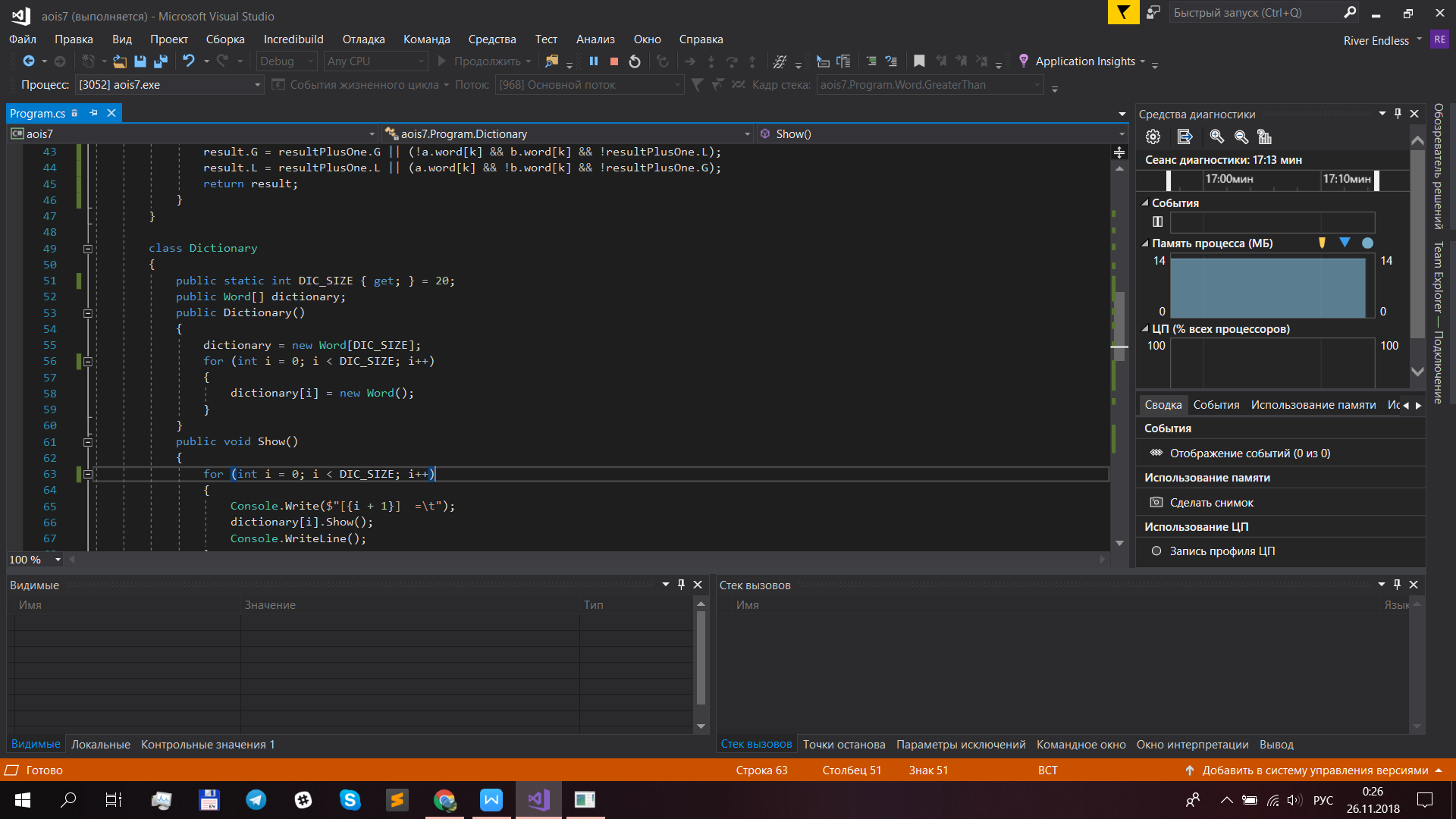
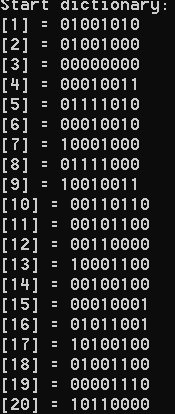
Построить и проверить программную модель ассоциативного процессора с применением рекуррентных алгоритмов – алгоритм поиска величин, заключенных в данном интервале, алгоритм поиска по соответствию.

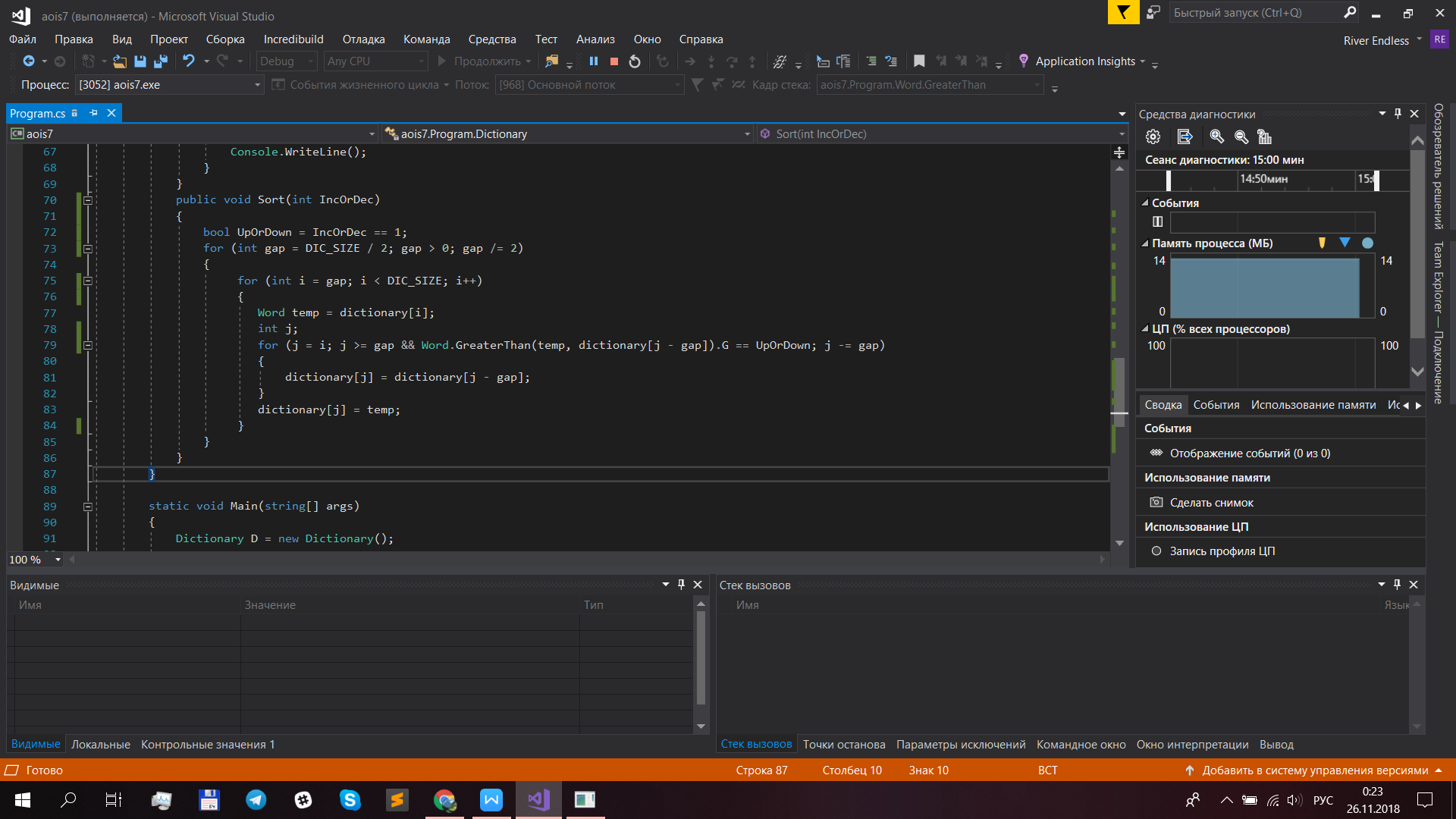
**Вариант 28**

Упорядоченная выборка (сортировка) - по убыванию и по возрастанию.

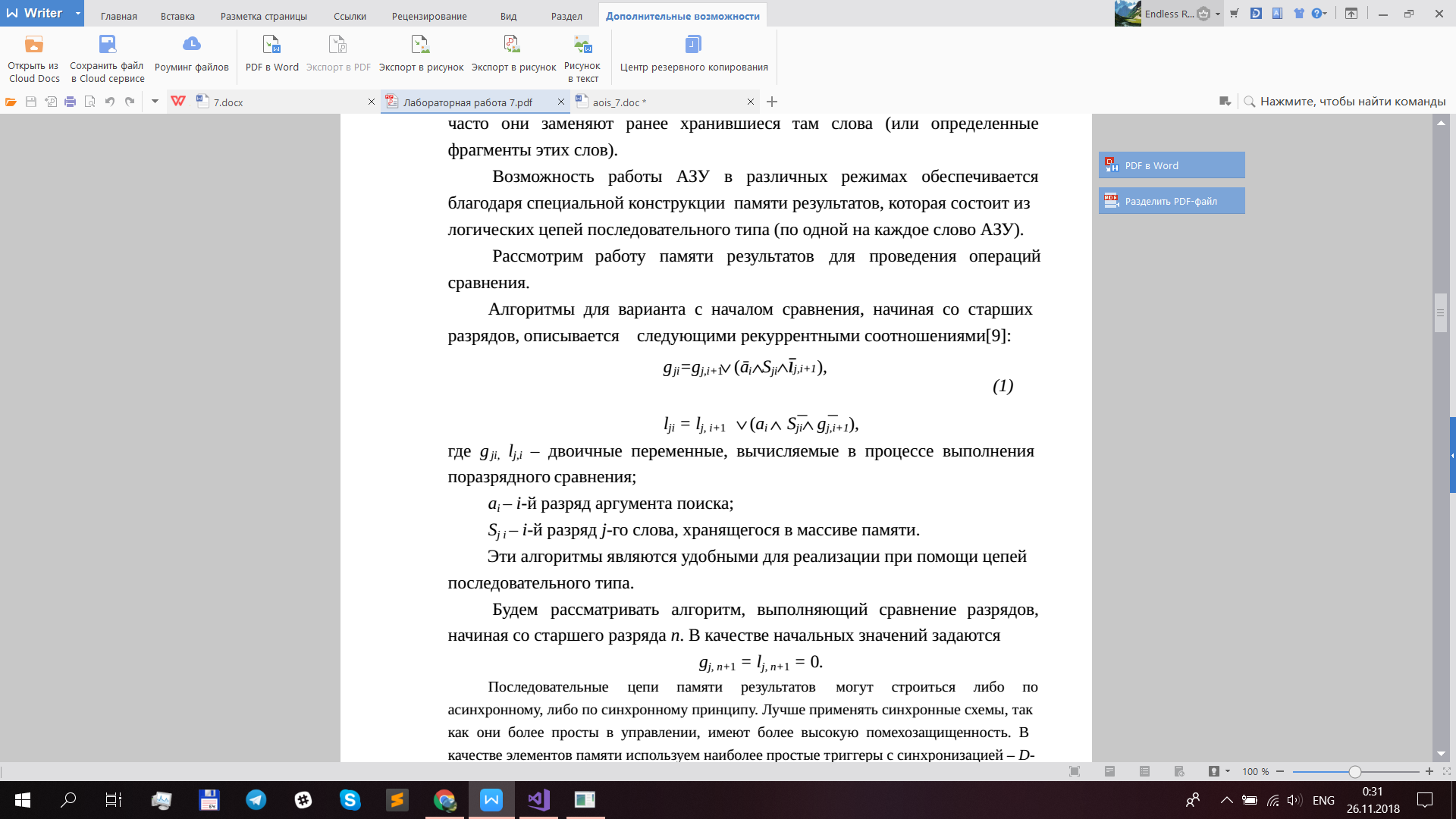
**Ход и результаты работы**

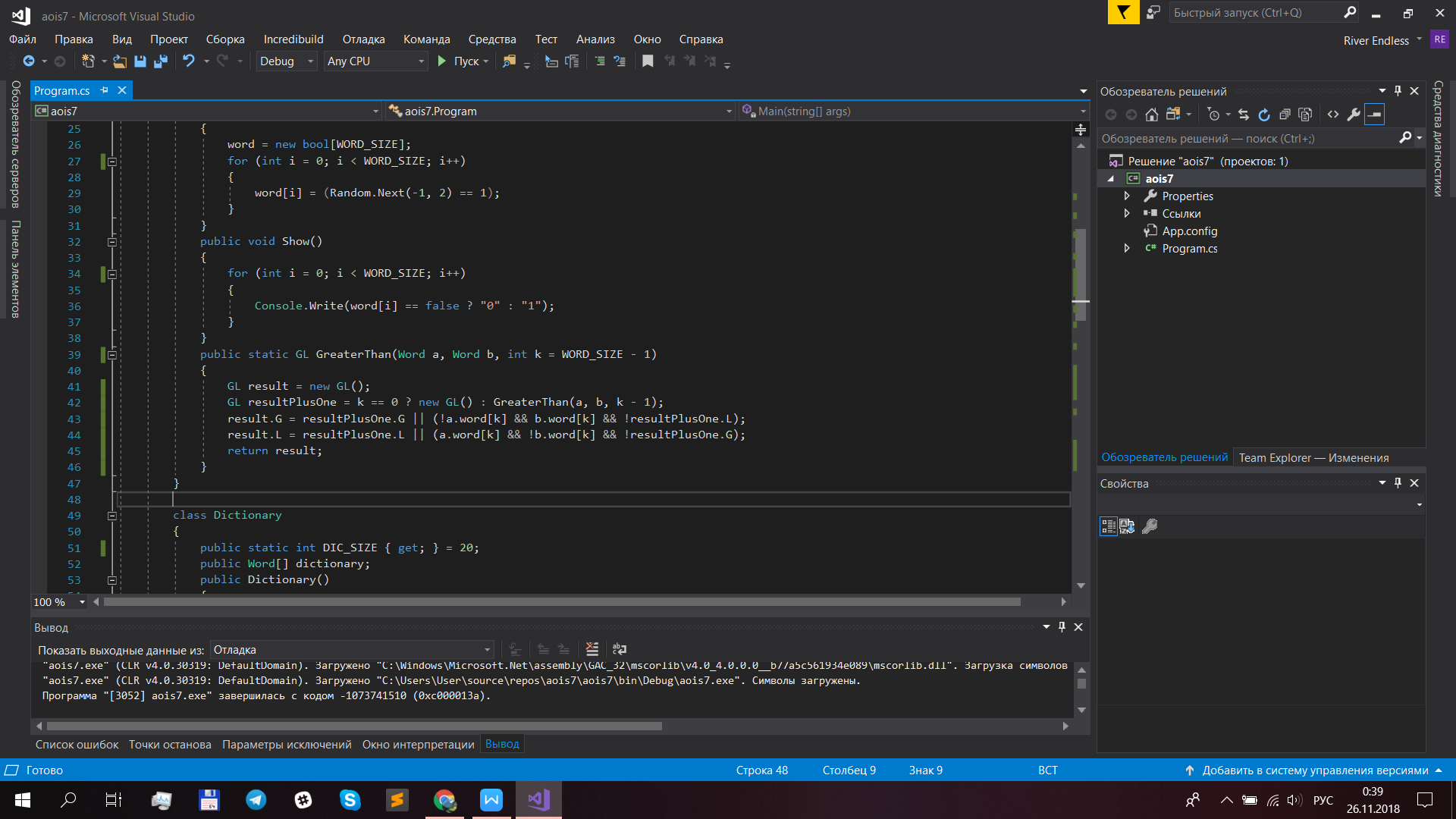
1. Проектируемый процессор будет работать над участком памяти, который представляет собой список слов - массив массивов (матрица) булевых переменных. Каждая строка-массив этой матрицы есть слово, а вся матрица - словарь. Количество слов в словаре - 20, количество символов в каждом слове - 8 (соответствующие значения заданы программно).

ъ

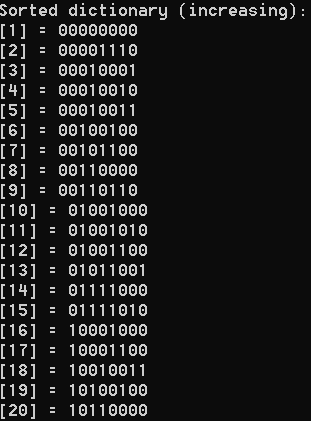
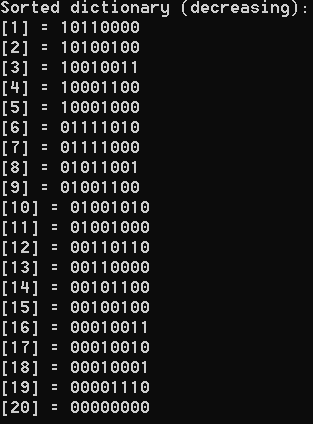
2. Отсортируем полученный словарь. В качестве оператора сравнения будем использовать рекуррентную функцию, которая сравнивает два слова и возвращает структуру, показывающую, которое из переданных в параметрах слово больше. Это осуществляется следующим способом:

Переменная IncOrDec указывает, будет сортировка по убыванию (значение 0) или по возрастанию (значение 1).

Метод GreaterThan() и является описанным рекуррентным алгоритмом. Его работа заключается в следующем: в параметры метода передаются два слова для сравнения. Начиная со старшего разряда, соответствующие разряды двух слов сравниваются, и, подчиняясь законам

мы формируем окончательную структуру GL, состоящую из двух булевых переменных: L (флаг «меньше» («less»)) и G (флаг «больше» («greater»)). Если первое слово в списке параметров больше второго, то G в возвращенной из метода структуре будет иметь значение 1; если меньше, то L будет равно 1; если слова равны, G = L = 0.

3. Полученные отсортированные массивы будут иметь вид:



**Вывод**

В прошлой лабораторной работе мы работали с хеш-таблицами, где данные упорядочивались с помощью системы «ключ-адрес»; при поиске информации в таких таблицах программа работает с адресом, а не с содержимым данных.

Понятие ассоциативного процессора подразумевает, что обработка ведется не над адресами, а над самим содержимым данных. Такие процессоры основаны на АЗУ - ассоциативном запоминающем устройстве, где доступ к данным осуществляется при помощи набора отличительных признаков. В ассоциативных процессорах, в отличие от АЗУ, также присутствуют методы обработки данных.

Эти методы - рекуррентные алгоритмы, работающие с отдельными ячейками памяти последовательно и по их разрядам - параллельно. Рекуррентные алгоритмы имеют шаги, которые зависят от предыдущих шагов. В случае сравнения двух слов на двух триггерах, именованных G и L, ассоциативного процессора проставляются значения 1 и 0 в соответствии с формулами, приведенными в реализованной программе. В итоге такое сравнение является ключевым в реализации более сложных алгоритмов: поиска, поиска по маске, поиска в заданном интервале, сортировке и т.д.