МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

Кафедра вычислительной техники

Лабораторная работа №3

по дисциплине «Технология программирования»

|  |  |
| --- | --- |
| Группа | АВТ-819 |
| Вариант | 2 |
| Студент | Ванин К.Е. |
| Преподаватель | Гаврилов К. В. |

Новосибирск 2020

**Задание:**

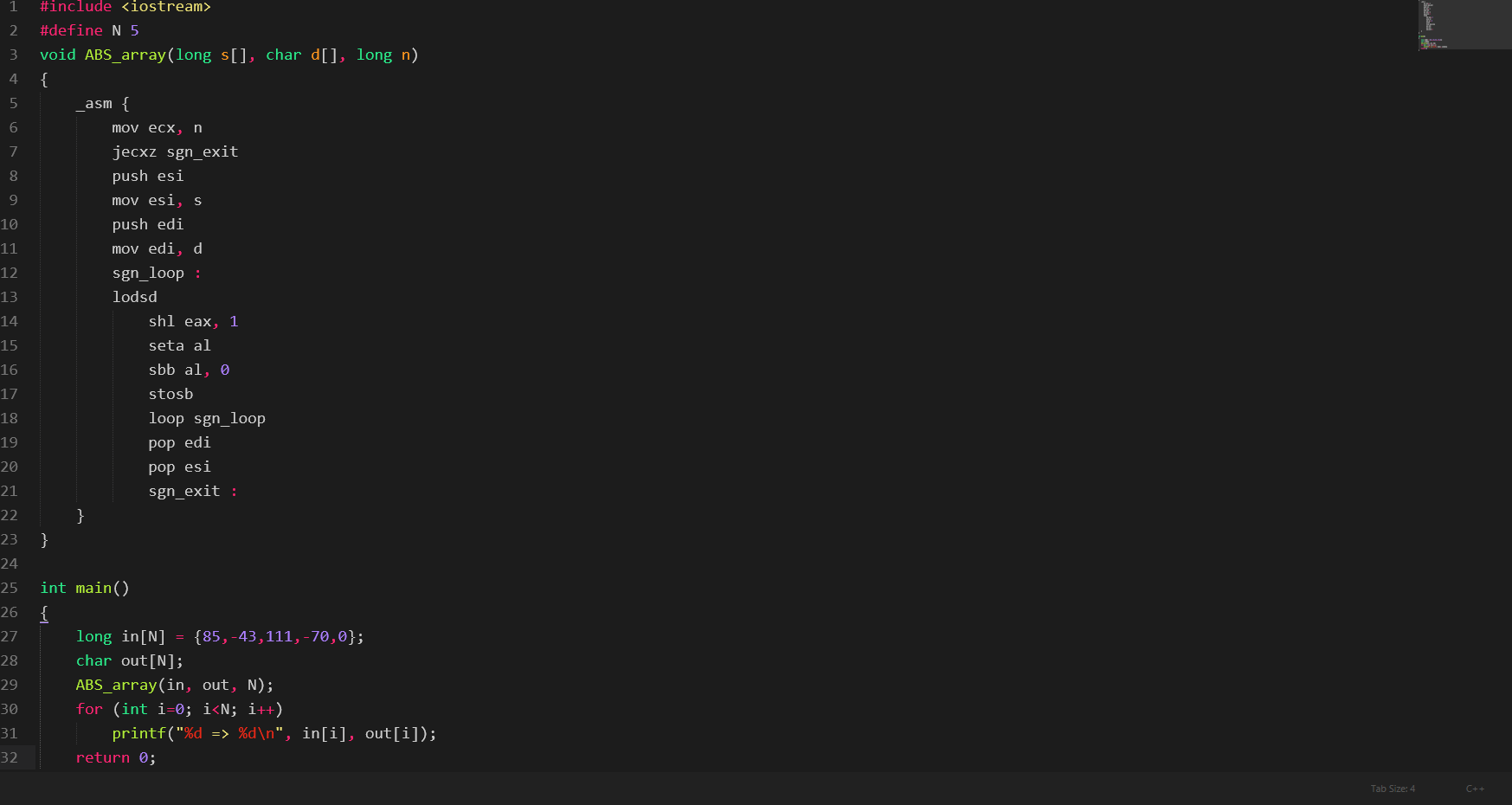
Вычисление функции сигнум. Тип элементов исходного массива — **long**, результирующего — **char** (одного байта достаточно для хранения результата функции **sgn**). Параметр s — исходный массив, d — результирующий, n — количество элементов.

1. Объявить два массива (исходный и результирующий), и исходный массив заполнить некоторыми целочисленными значениями.

2. Вызвать для данных массивов функцию, которая написана в виде ассемблерной вставки.

3. Вывести результирующий массив на экран.

**Листинг:**

****

**Результат выполнения программы:**

****

**Работа алгоритма:**

1. Нужно занести длину массива в регистр **ecx** (счётчик цикла)
2. Если длина массива = 0 => переходим по метке и работа программы завершается.
3. Команда **loop** в первую очередь уменьшает счетчик, а потом проверяет, равен ли он нулю, поэтому на нуль также следует проверить.
4. Заносим в машинный стек чтобы вернуть их на место после выполнения нашей задачи.

1) значение **esi**

2) значение **edi**

1. В регистр **esi** отправляется адрес нулевого элемента входного массива, в регистр **edi** – выходного массива.
2. Команда **lodsd** считывает двойное слово по адресу из регистра **esi** в регистр **eax**. Значение регистра **esi** увеличивается на 4 (т.е. будет адрес следующего элемента массива). Флаг **DF** = 0 по умолчанию, поэтому значение увеличивается.
3. Значение в **eax** сдвигается влево на 1 разряд (умножаем на 2) и заносится обратно в **eax**, при этом старший (знаковый) бит заносится во флаг **CF**. Если новое значение **eax** = 0, то **ZF** = 1.
4. Если **CF** = 0 и **ZF** = 0 (число положительно), то устанавливается младший байт **eax** – **al**, который равен 1. В ином случае (число отрицательное) – нуль.
5. С помощью целочисленного вычитания с займом вычитаем из **al** значение флага **CF**.

Если число положит. **al** = 1 – 0 = 1, если отриц. **al** = 0 – 1 = -1, если число равно нулю **al** = 0 – 0 = 0.

1. Команда **stosb** записывает содержимое **al** по адресу, который находится в регистре **edi** (выходной массив). После этого значение регистра **edi** повышается на 1.
2. Команда **loop** образует цикл: уменьшаем значение регистра **ecx** на 1, после чего, если значение регистра **ecx** != 0 => переходим к 5 пункту. Если значение **ecx** = 0, то обработаны все элементы массива, выполняется следующая команда.
3. Извлекаем из стека и помещаем обратно сначала значение **edi**, которое сохраняется в самом начале, после значение **esi**.