Косинов Денис БПИ239.

Вариант 33.

АВС ИДЗ#1.

**Код программы:**

<https://github.com/o7Techno/CSA-HW/tree/main/%D0%90%D0%92%D0%A1%20%D0%98%D0%94%D0%97%231>

**Описание программы:**

Программа разбита на 3 файла: IDZ1.ams выступает main файлом, там запускается сама программа и оттуда идут вызовы дальнейших функций (Так же там находится exit (он 1 с целью повысить читабельность кода, не дублируя выход для разных режимов выполнения программы) (нечто напоминающее return 0; на c++)); functions.asm содержит в себе все подпрограммы вызываемые в программе; macros\_lib.inc - библиотека макро.

**Описание подпрограмм:**

print\_int - выводит на дисплей int значение находящееся в регистре a0.

print\_string - выводит на дисплей строковое значение находящееся в регистре a0.

read\_int - считывает int значение с клавиатуры и возвращает его в регистр a0.

input - считывает с клавиатуры и записывает находящееся в a0 количество int значений в область памяти начинающуюся с адреса в a1. (Важно, что значение в a0 не должно превышать 10 и быть меньше 2)

output - выводит на дисплей находящееся в a0 количество int значений из памяти начиная с адреса в a1.

array\_creator - заполняет область памяти int значениями следующим образом:

количество элементов подается в регистр a0, адрес начала “массива” подается в регистр a2, данные для записи получаются так new[i] = old[i] - old[i + 1], где new[i] - i-тое значение нового массива, а old[i] - i-тое значение старого массива; Адрес начала старого массива подается в регистр a1.

check\_arrays - на вход получает адрес начала первого массива в регистре a0, адрес начала второго массива в регистре a1, адрес конца первого массива в регистре a2, и если все элементы этих двух массивов равны выводит на дисплей “Everything is correct” или, если хотя бы 2 соответствующих элемента (с одним индексом) не совпадают выводит на дисплей “Incorrect at index [index]”.

**Описание макро:**

input\_macro - обертка над подпрограммой input, которая получает количество элементов массива для записи с клавиатуры, проверяет корректность полученных данных (кол-во данных массива должно быть [2; 10] для корректной работы программы) и совершает ввод и запись данных в статическую область памяти данных с клавиатуры (массива).

array\_creator\_macro - обертка над подпрограммой array\_creator, которая получает данные для вызова функции output и вызывает ее.

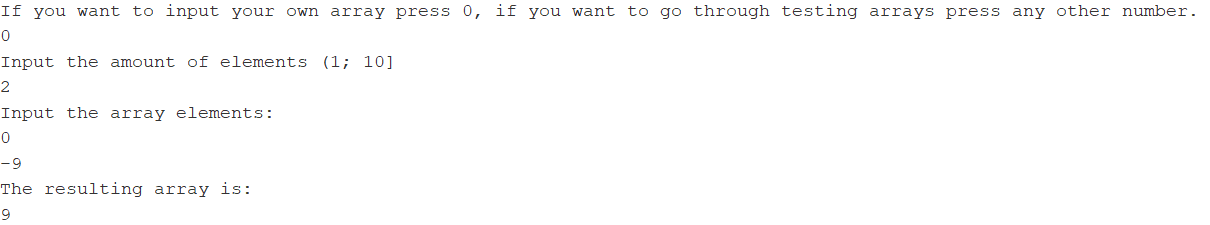
output\_macro - обертка над подпрограммой output, которая получает данные для вызова функции output и вызывает ее.

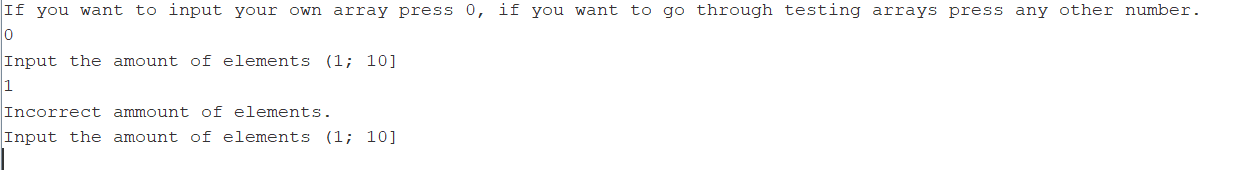
mode - макро, который в зависимости от введенного режима пользования программой (свои данные или автоматическое тестирование) выполняет соответствующий функционал. Т.е. является самой крупной по вложению оберткой исполнения функционала программы.

**Как использовать программу:**

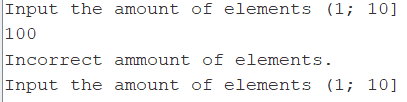
Первый ввод пользователя - режим, в котором он хочет использовать данную программу. (0 - ввод данных самому, любое другое число - автотестирование). Далее, если пользователь ввел не 0, то программа выводит, корректно ли было завершено автотестирование (все ли ответы программы, совпали с действительными ответами) и на этом программа завершается, если же пользователь ввел 0 на выборе режима, то следующий его ввод будет означать количество элементов в массиве - целое число от 2 до 10 включительно (есть проверка на это), после чего пользователю необходимо ввести данное количество целых чисел. Программа выведет полученный обработанный массив (согласно с условием варианта).

**Тесты (руками):**

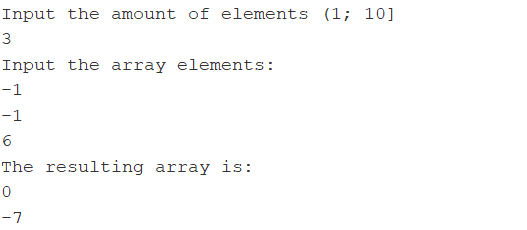
****

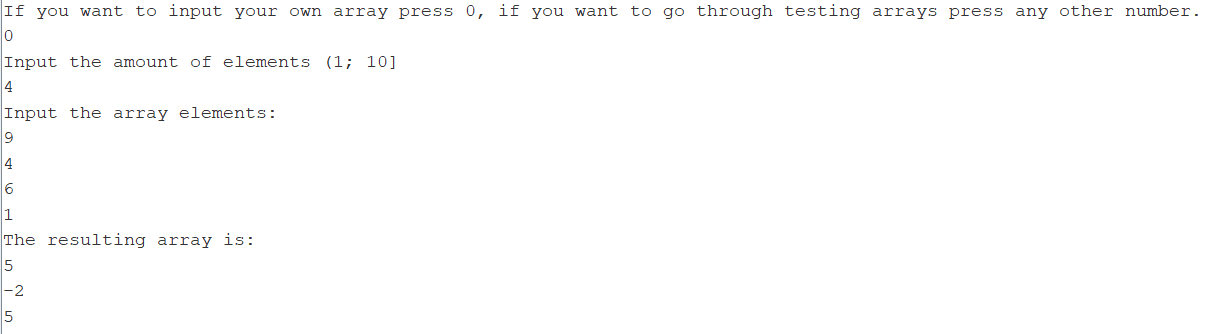
****

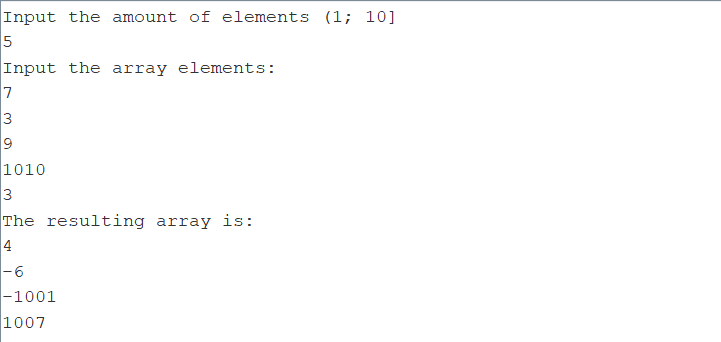
(было дано некорректное кол-во элементов массива (меньше требуемого) спрашивается еще раз)

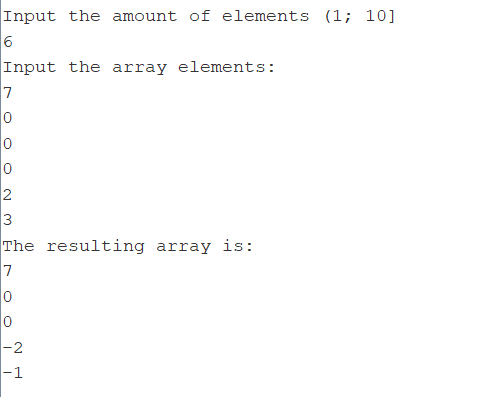


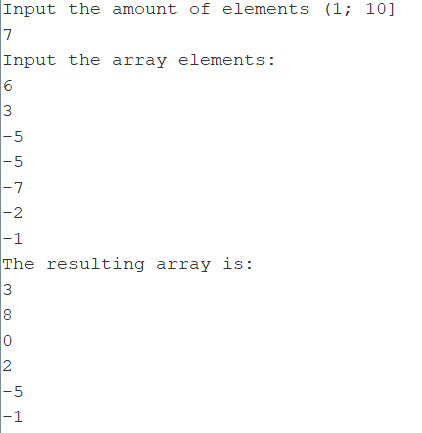
(было дано некорректное кол-во элементов массива (больше требуемого) спрашивается еще раз)

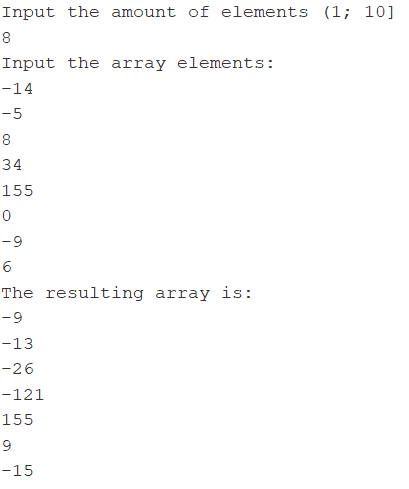


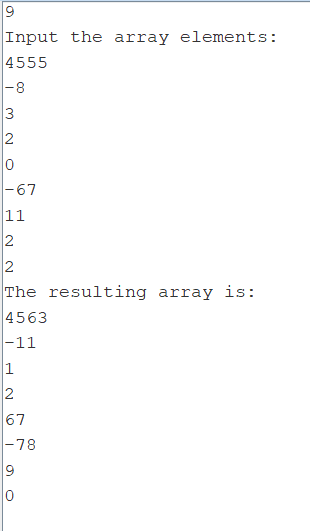


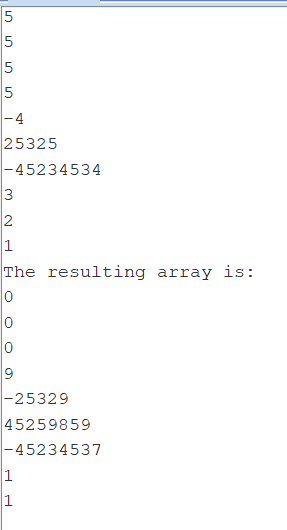












**На 6-7 баллов:**

Все функции получают все необходимые данные (если они необходимы) через регистры a0 - a7, используют только регистры t (стек не используется), а также внутри функций не используются данные не переданные им (т.е. они атомарны). Что делает их соответствующими конвенции. Все комментарии присутствуют в коде.

**На 8 баллов:**

Т.к. функции поддерживают работу с формальными данными, они поддерживают многократное использование с различными входными данными. Реализовано автотестирование (подробнее см. следующий пункт).

**Как работает автоматическое тестирование:**

В .data с метки test\_begin до метки test\_end располагаются тестовые данные таким образом: .word <длина массива> <данные массива>

для 10 различных массивов. С метки test\_ans\_begin до метки test\_ans\_end располагаются вручную подсчитанные результаты обработки массива описанной в варианте. С метки test\_res зарезервирована память для результатов прогона программы по данным от test\_begin до test\_end. Когда пользователь вводит не 0,

программа вызывает array\_creator для каждого массива из промежутка от test\_begin до test\_end и записывает последовательно результаты выполнения начиная с метки test\_res. После чего вызывается функция check\_arrays, которая проверяет совпадают ли данные в массивах с test\_ans\_begin до test\_ans\_end с данными полученными в результате вызовов array\_creator. Если данные совподают, то выводится “Everything is correct”, если какая-то пара не совпадает выводится “Incorrect at [index]”.

**На 9 баллов:**

Выполнены требования по использованию макросов - оберток (подробнее см. описание макро).

**На 10 баллов:**

Программа разбита на 3 различных ассемблерных файла (подробнее см. описание программы). Подпрограммы print\_string, print\_int, read\_int - унифицированные модули, которые вызываются повторно в обоих режимах работы программы.

Макро лежат в библиотеке макро в отдельном файле, подпрограммы лежат в отдельном ассемблерном файле.