Дмитриев Дмитрий Игоревич, БПИ227

**Вариант 25. Условие задачи:** Сформировать массив B из элементов массива A заменив все положительные числа значением 2, а отрицательные — увеличить на 5.

**Претендую на 10 баллов за выполнение задания**

[**Репозиторий с кодом на Ассемблере и автоматическими тестами**](https://github.com/TIN-slayer/AVS_IDZ-1)

(Для запуска основной программы нужно использовать main.asm, а для тестов tests.asm)

**Прогрессирующий отчёт на баллы от 4 до 10:**

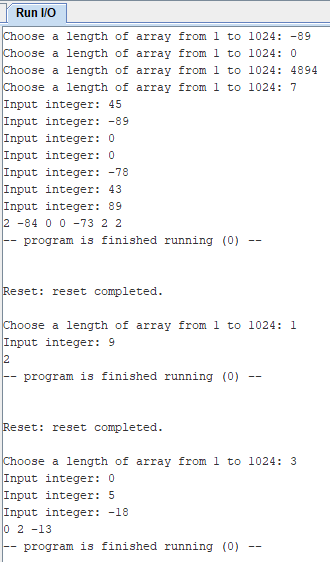
**4-5 баллов:**

1. Решил задачу 25 на Ассемблере. Оформил ввод с клавиатуры вывод в консоль.
2. Комментарии добавил.
3. Реализовал подпрограммы ввода, формирования нового массива по данному в условии правилу и вывода:

(**Замечание:** для экономии памяти я формирую массив B там, где хранится A, по сути, меняя массив A на B, поэтому в дальнейшем этот процесс я буду называть изменением)

* 1. Подпрограмма ввода (**input**) принимает на вход адрес начала массива в регистре a0, запрашивает у пользователя длину массива и добавляет в него новые элементы (ввод происходит с клавиатуры). Возвращает длину массива в регистре a1.
  2. Подпрограмма изменения массива (**logic**) принимает на вход адрес начала массива и его длину в регистрах a0 и a1 соответственно. Заменяет положительные элементы на 2, а к отрицательным добавляет 5 (число 0 остаётся прежним).
  3. Подпрограмма вывода (**output**) принимает на вход адрес начала массива и его длину в регистрах a0 и a1 соответственно. Выводит в консоль измененный массив.

1. Прикладываю результаты **ручных** тестовых прогонов:

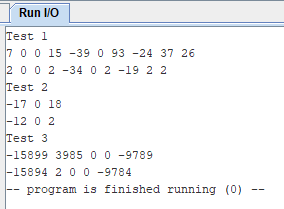


**6-7 баллов:**

1. В tg чате АВС Александр Иванович писал, что, если регистров (a0-a7) хватает, стек использовать не надо. Поэтому аргументы в стек я не сохранял, но регистр адреса возврата по конвенции сохранял во всех подпрограммах.
2. По аналогичной причине я не использовал локальные переменные, а только регистры-аргументы и временные.
3. Добавил комментарии для принимаемых и возвращаемых аргументов подпрограмм(функций).

**8 баллов:**

1. Подпрограммы input, logic и output независимы от main и могут принимать аргументы от другой программы с другими массивами (Например, у меня реализована программа tests.asm для автоматического тестирования).
2. Реализовал программу **tests.asm** для автоматического тестирования на разных массивах. Результат **автоматических** тестовых прогонов:



**9 баллов:**

1. Реализовал макросы ввода и вывода (print\_int, read\_int, print\_str, print\_char, **input**, **output**), поддерживающие различные аргументы.
2. Обернул подпрограммы input, logic, output одноимёнными макросами.

**10 баллов:**

1. Разбил программу на несколько файлов, которые собираются вместе во время компиляции. Подпрограммы logic и output используются и в main.asm, и в tests.asm, также и там, и там используются макросы из общей библиотеки.
2. Выделил макросы в автономную библиотеку macroslib.asm.