

## ACS. Индивидуальное Домашнее Задание 1

Горбачева Маргарита Валерьевна | БПИ-245.

### Задание

Разработать программу, в которой вводится одномерный массив  $A$ , состоящий из  $N$  элементов (значение  $N$  вводится в диалоге с пользователем при выполнении программы). После этого из элементов массива  $A$  формируется новый массив  $B$  в соответствии с условиями, указанными в варианте, и осуществляется его вывод. Память под массивы может выделяться статически, на стеке, автоматически по выбору разработчика с учетом требований к оценке работы. При решении задачи необходимо использовать подпрограммы для реализации ввода, вывода и формирования нового массива массива. Допустимы (при необходимости) дополнительные подпрограммы. Максимальное количество элементов в массиве не должно превышать 10 (ограничение должно контролироваться при вводе данных). Необходимо обрабатывать некорректные значения как для нижней, так и для верхней границ массивов в зависимости от условия задачи.

### Вариант 16

Сформировать массив  $B$  из элементов массива  $A$ , заменой значений, которые больше среднего арифметического, значением среднего арифметического массива  $A$ .

### Решение, претендующее на 8 баллов

Файлы программы хранятся на GitHub по ссылке: <https://github.com/Misss-Lacoste/ACS>

### Общий алгоритм:

Наша задача - из массива  $A$  собрать массив  $B$  таким образом, что если эл-ты массива  $A$  меньше среднего арифметического из чисел всего массива  $A$ , то в массив  $B$  такие элементы копируются, какие они есть. Если же значения массива  $A$  больше или равны(!!) среднему арифметическому, то в массив  $B$  вместо них записывается среднее арифметическое массива  $A$ .

### Подробные шаги:

- 1) Вычисляем среднее арифметическое. Изначальная сумма  $= 0$ , далее сумма  $=$   $=$   $\text{изначальная сумма} + \text{новый рассматриваемый элемент}$ ; в завершении,  $\text{среднее арифметическое} = \text{сумма} / \text{кол-во эл-тов}$ .
- 2) Формируем массив  $B$ :

Для каждого элемента в массиве A: если элемент < среднего арифм-кого, то необходимо записать элемент в массив B. Иначе: записать среднее арифм-кое в массив B.

В реализации данного задания принимают участие **6 файлов** (GitHub):

1. `main.asm` - основной исполняемый файл, точка входа в программу. Содержит функцию `main` - старт программы, логику вызова всех подпрограмм в правильной последовательности, интеграцию всех модулей в единую программу.
2. `test.asm` - автономное тестирование подпрограмм. содержит функцию `test_main` (чтобы не конфликтовать с основной программой, заранее подготовленные тестовые данные, проверку корректности работы математической функции вычисления нашего среднего арифметического для заполнения массива, а также вывод результатов тестирования для проверки).
3. `file_io.asm` - отвечает за чтение и запись данных в файлы, содержит подпрограммы: `read_from_file` - чтение массива из файла, а также обработку файловых дескрипторов и ошибок.
4. `io.asm` - обеспечивает взаимодействие с пользователем через консоль. Содержит подпрограммы: `input_int` - ввод одного числа и `input_array` - ввод массива с клавиатуры, а также `print_array` - красивый вывод массива в формате `[1, 2, 3]`, форматированные сообщения и приглашения ко взаимодействию в консоли.
5. `math.asm` - отвечает за вычислительную логику задачи. Содержит подпрограммы: `calculate_avg` - вычисление среднего арифметического и `form_array_B` - формирование результирующего массива по условию. В файле используются преобразования между целыми и вещественными числами, сравнения и условные операции.
6. `text.txt` - файл с входными данными для тестирования.

#### Тесты:

- 1) Обычный примитивный случай со входными данными : `[1, 2, 3, 4, 5]` - проверяем базовую функцию алгоритма.
- 2) Все элементы больше среднего: `[10, 20, 30]`
- 3) Один элемент: `[5]`, проверяем корректность при `size = 1`.
- 4) Все элементы одинаковые: `[3, 3, 3]`.
- 5) Тест с отрицательными числами `[-1, -2, 6]`.
- 6) Самое интересное - дробное среднее арифметическое на наборе `[1, 2]`.  
Проверяем округление при преобразовании `float` → `int`, потерю точности при конвертации типов, поведение с нецелыми средними значениями.

## Результаты автоматических тестов:

```
MENU
Enter '0': input your array.
Enter other symbol: launch autotests.
```

```
Your choice: 3
ТЕСТ 1: [1,2,3,4,5] -> [1,2,3,3,3]
[1, 2, 3, 4, 5]
[1, 2, 3, 3, 3]
√Тест пройден успешно
```

```
ТЕСТ 2: [10,20,30] -> [10,20,20]
[10, 20, 30]
[10, 20, 20]
√Тест пройден успешно
```

```
ТЕСТ 3: [5,5,5,5] -> [5,5,5,5]
[5, 5, 5, 5]
[5, 5, 5, 5]
√Тест пройден успешно
```

## Если мы тестируем программу самостоятельным вводом:

```
MENU
Enter '0': input your array.
Enter other symbol: launch autotests.
```

```
Your choice: 0
Выберите способ ввода:
1 - Клавиатура
2 - Файл
```

```
Введите выбор: 1
Введите количество элементов массива (1-10): 3
A[0] = 2
A[1] = -8
A[2] = 0
Массив A: [2, -8, 0]
Среднее арифметическое: -2.0
Массив B: [-2, -8, -2]
If you want to finish program, enter '0', otherwise press any other number to restart: |
```

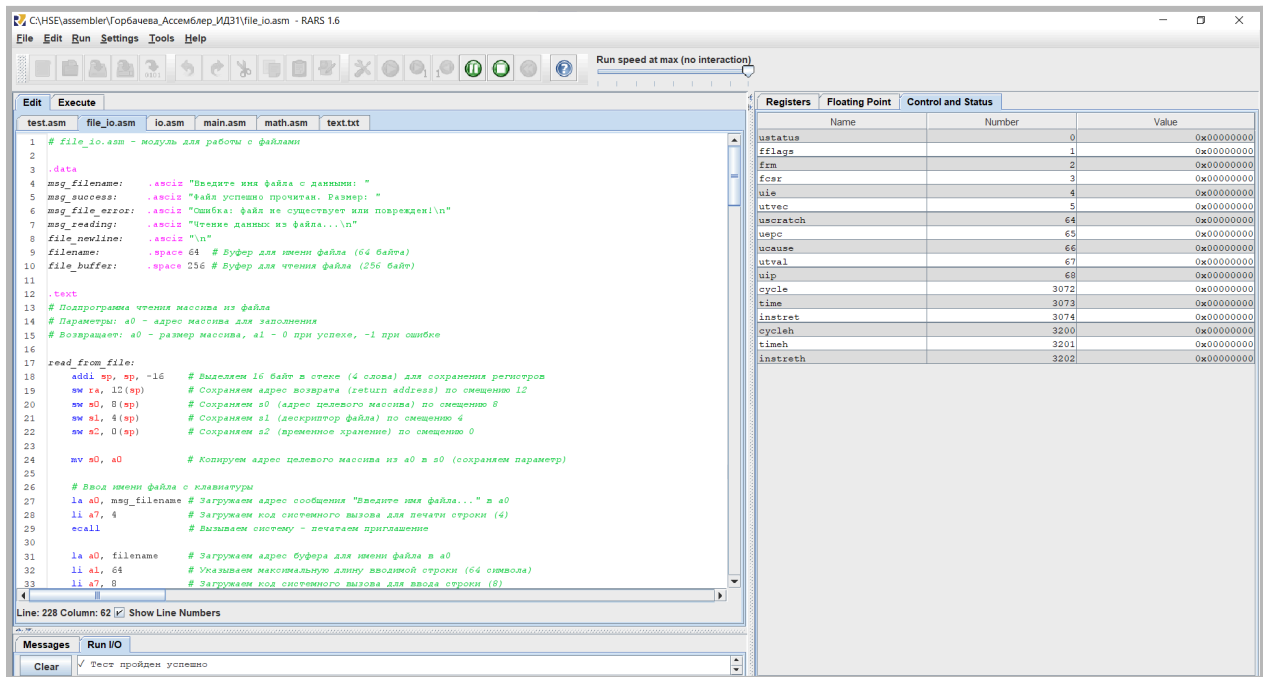
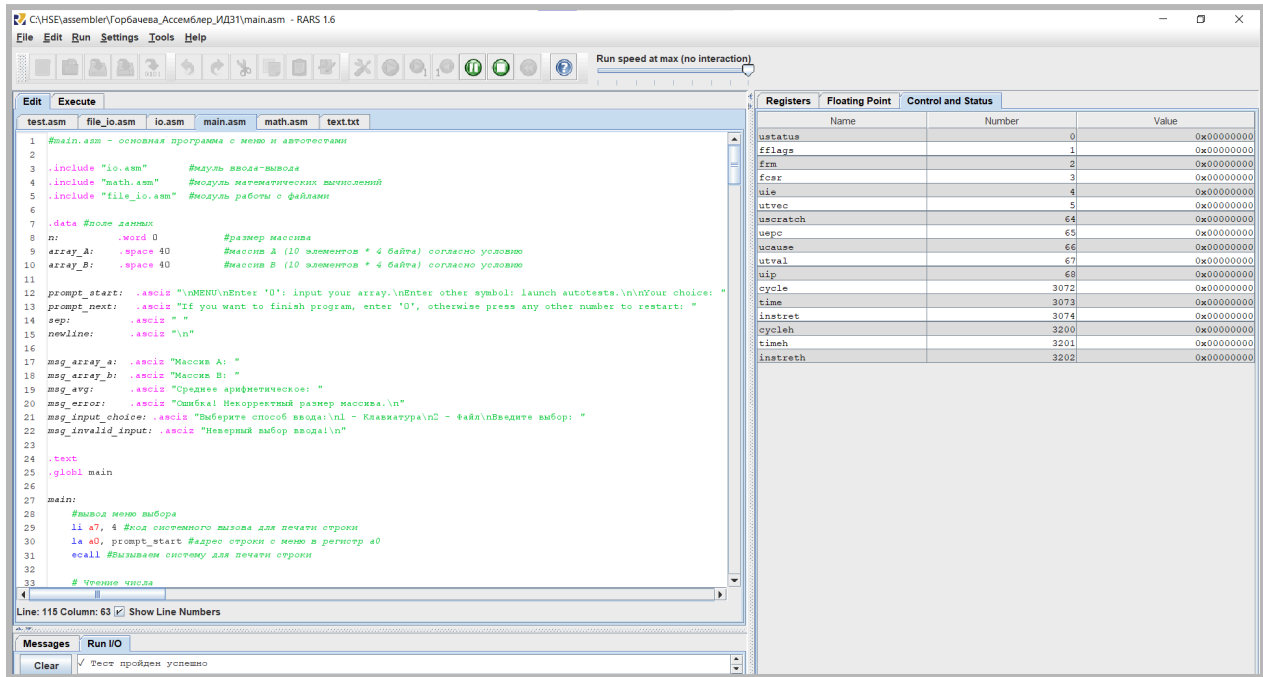
```
Введите выбор: 2
Введите имя файла: text.txt
Ошибка открытия файла!
Ошибка! Некорректный размер массива.
Массив A: []
Среднее арифметическое: NaN
Массив B: []
If you want to finish program, enter '0', otherwise press any other number to restart: |
```

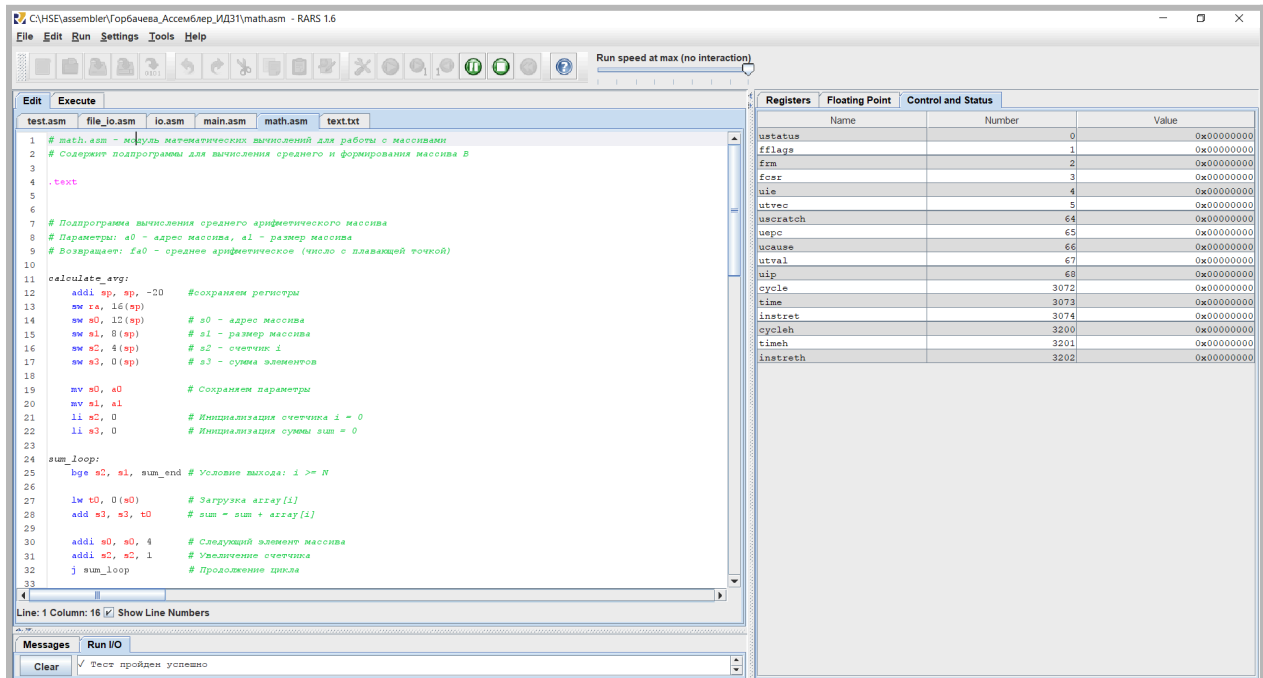
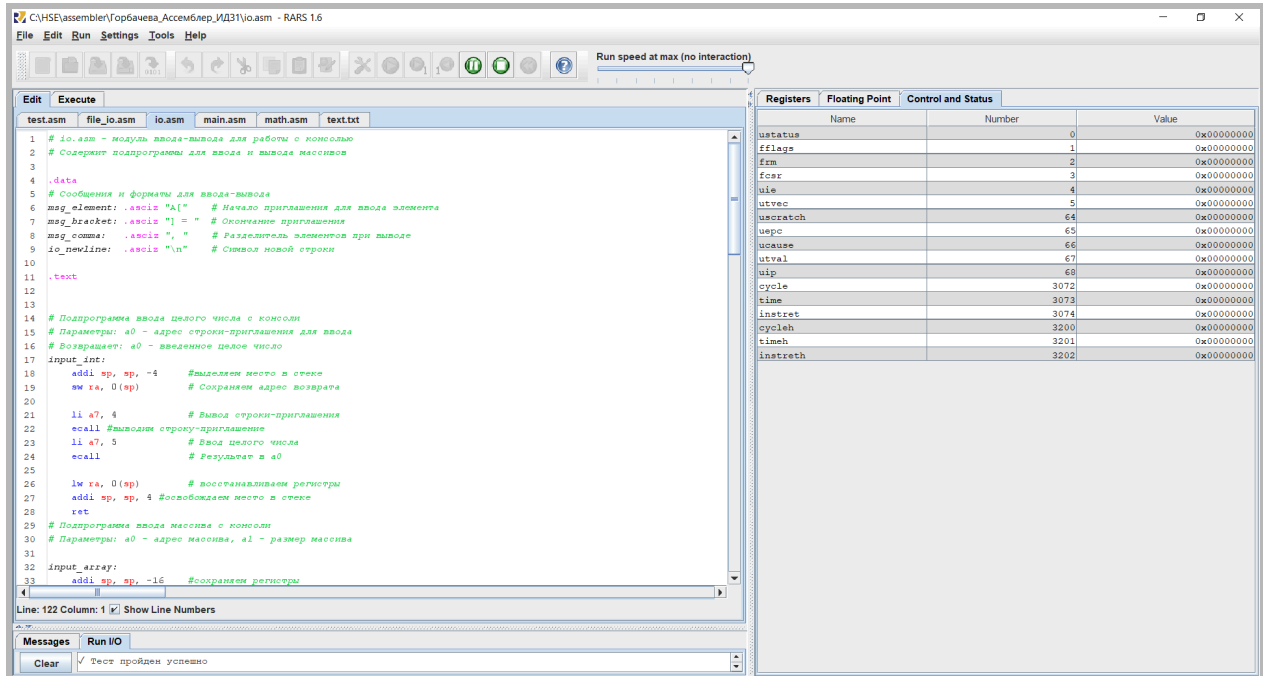
```
Your choice: 0
Выберите способ ввода:
1 - Клавиатура
2 - файл
Введите выбор: 1
Введите количество элементов массива (1-10): 8
A[0] = 1
A[1] = 0
A[2] = -4
```

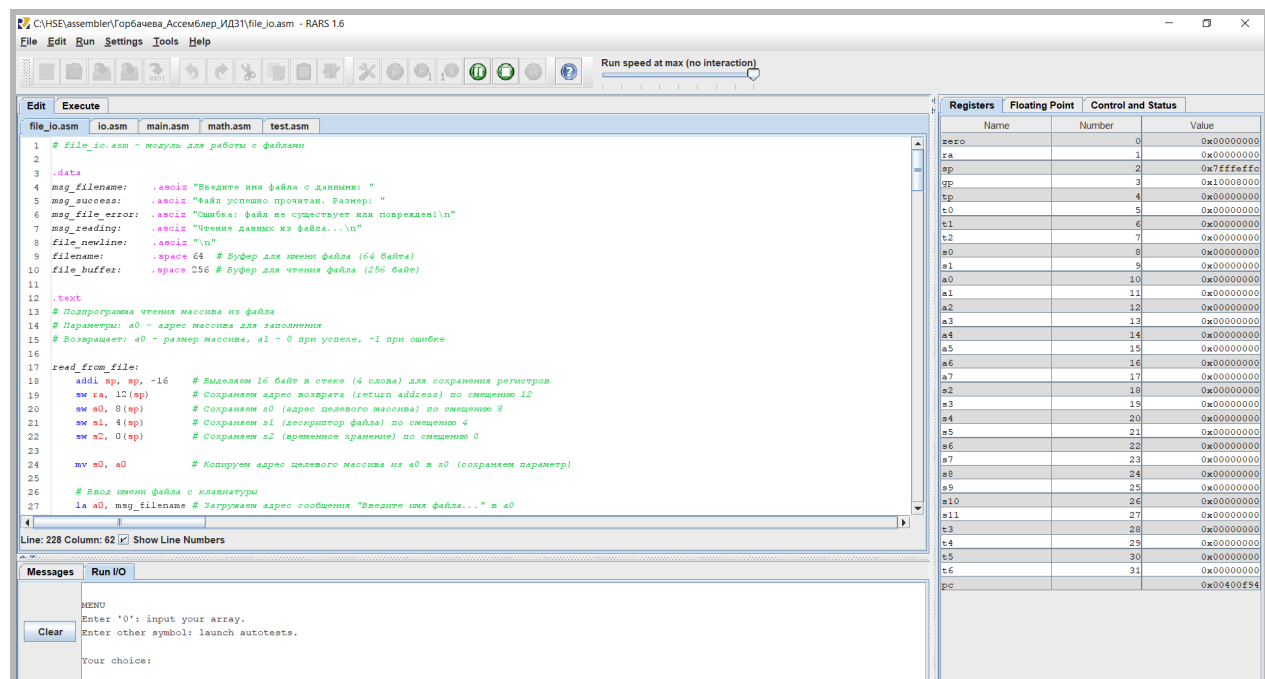
```
A[0] = 1
A[1] = 0
A[2] = -4
A[3] = 3
A[4] = 6
A[5] = 7
A[6] = 2
A[7] = -9
Массив A: [1, 0, -4, 3, 6, 7, 2, -9]
```

```
Массив A: [1, 0, -4, 3, 6, 7, 2, -9]
Среднее арифметическое: 0.75
Массив B: [1, 0, -4, 1, 1, 1, 1, -9]
If you want to finish program, enter '0', otherwise press any other number to restart: |
```

## Скриншоты некоторых файлов:







## Выполненные условия:

На 4-5:

1. Есть решение на ассемблере. Есть ввод и вывод данных на экран.
2. Есть комментарии.
3. Использование подпрограмм без параметров и локальных переменных.
4. Отчёт с полным тестовым покрытием.

На 6-7:

5. В программе необходимо использовать подпрограммы с передачей аргументов через параметры, отображаемые на стек.
  6. Использование подпрограмм с передачей аргументов через регистры по конвенции.
- В коде не понадобились такие аргументы, для всех подпрограмм они не нужны.
7. Сохранение локальных переменных в свободных регистрах.
  8. Комментарии к функциям.
  9. Изменения в отчёте. Решение сразу написано.

На 8:

10. Многократное использование подпрограмм.
11. Реализована дополнительная тестовая программа с автоматическими тестами (tests.asm).
12. Добавлена информация в отчёт.

/\*На 9, к сожалению, у меня не получилось реализовать макросы, однако из пункта на 10 чтение из файла реализовано\*/

На 10

13. Вместо ввода данных с клавиатуры использовать ввод размера массива и его элементов из файла. Имя файла с исходными данными вводить с использованием клавиатуры.