

Выполнил: Радионов Никита Дмитриевич БПИ236

Вариант 41

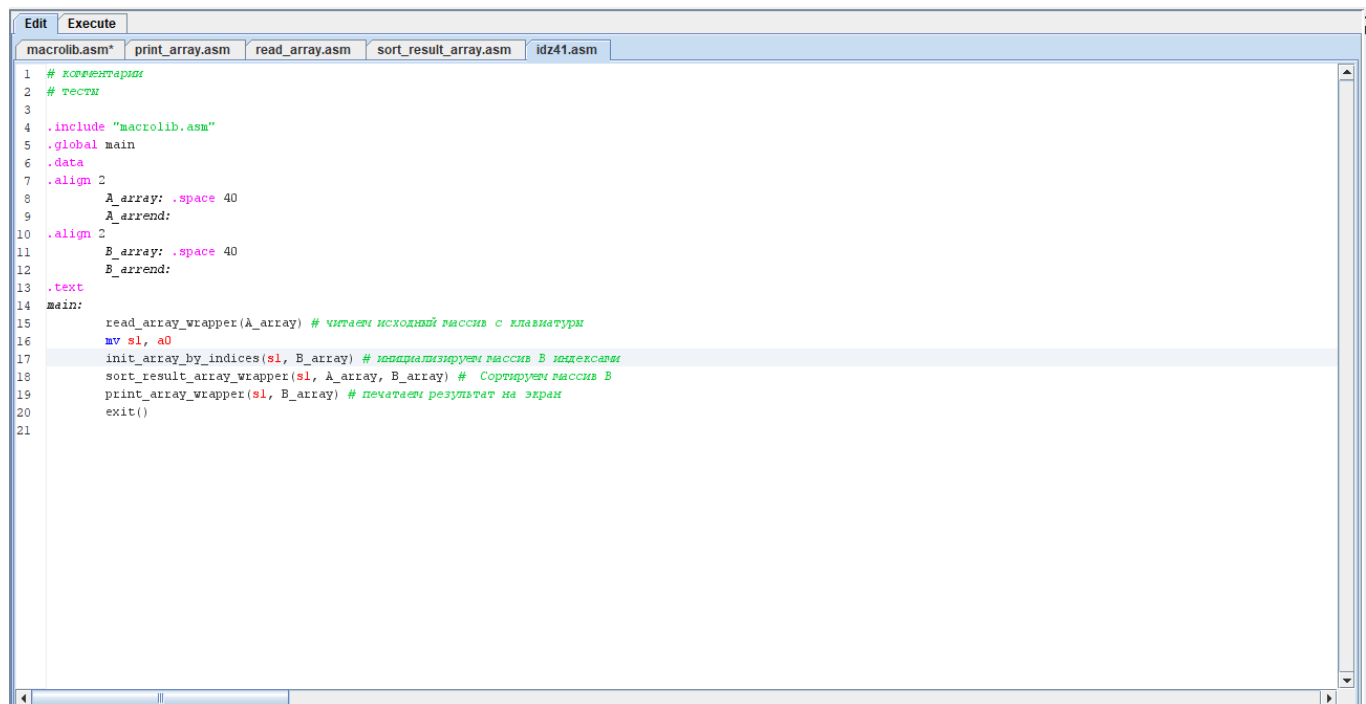
В моём варианте необходимо сформировать массив B из индексов элементов массива A . Порядок следования индексов в массиве B позволяет выводить элементы массива A по возрастанию.

Основной алгоритм:

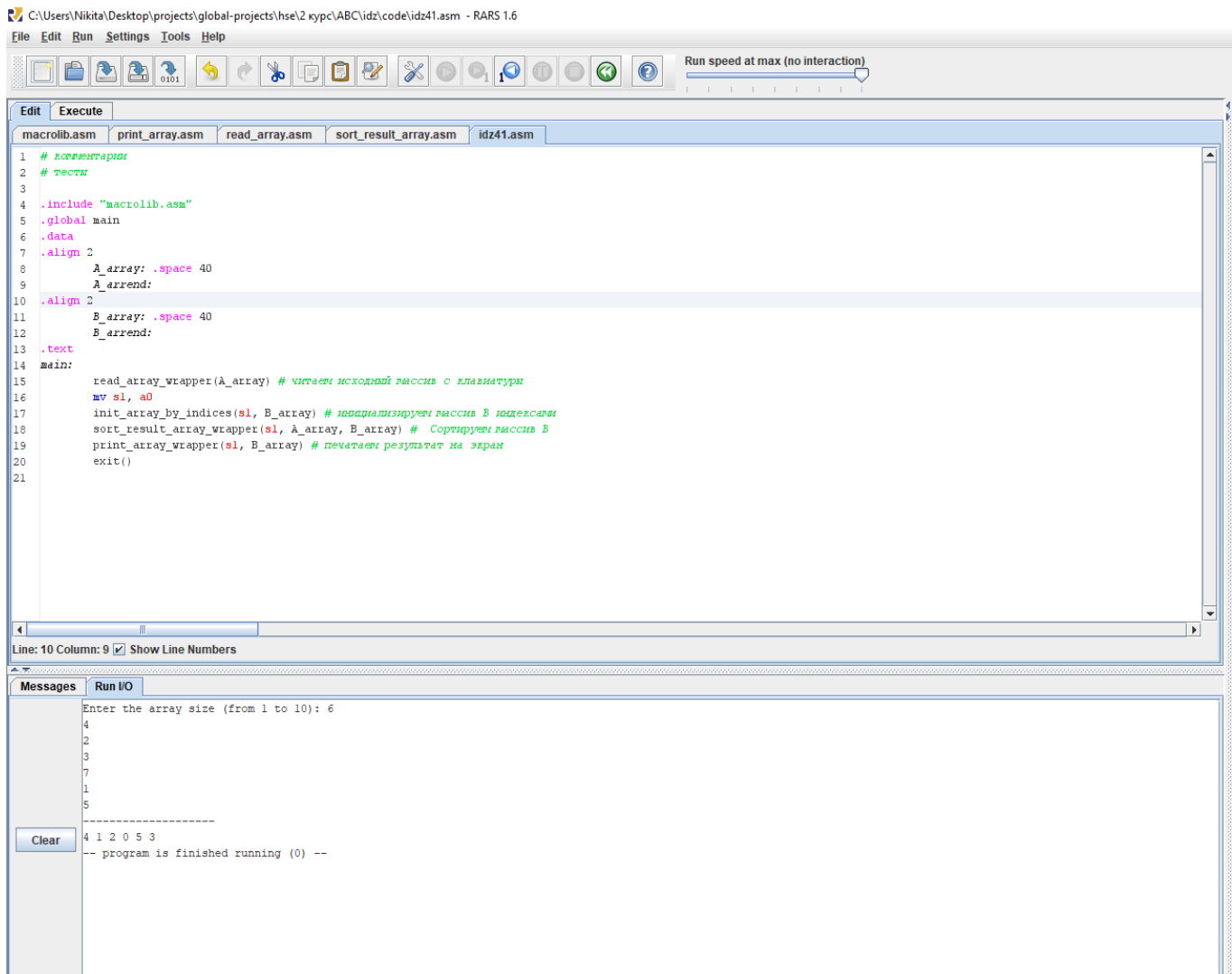
1. Составляем массив B из индексов (чисел от 0 до $n - 1$)
2. Сортируем массив B сортировкой пузырьком, при сравнении элементов используем элементы массива A

[Исходный код проекта](#)

Скриншот кода основной программы



```
1  # комментарии
2  # тесты
3
4  .include "macrolib.asm"
5  .global main
6  .data
7  .align 2
8      A_array: .space 40
9      A_arrend:
10 .align 2
11     B_array: .space 40
12     B_arrend:
13 .text
14 main:
15     read_array_wrapper(A_array) # читаем исходный массив с клавиатуры
16     mv $1, a0
17     init_array_by_indices($1, B_array) # инициализируем массив B индексами
18     sort_result_array_wrapper($1, A_array, B_array) # Сортируем массив B
19     print_array_wrapper($1, B_array) # печатаем результат на экран
20     exit()
21
```



Пройдемся по каждому требованию и покажем, что оно выполняется.

Требования на оценку 4-5:

№ 1

Приведено решение задачи на ассемблере. Ввод данных осуществляется с клавиатуры. Вывод данных осуществляется на дисплей.

Выполнение этого требования доказывает скриншот выше.

№ 2

В программе должны присутствовать комментарии, поясняющие выполняемые действия.

Как можно видеть со скриншота выше в основной программе комментарии присутствуют, как и в остальных частях многомодульного проекта.

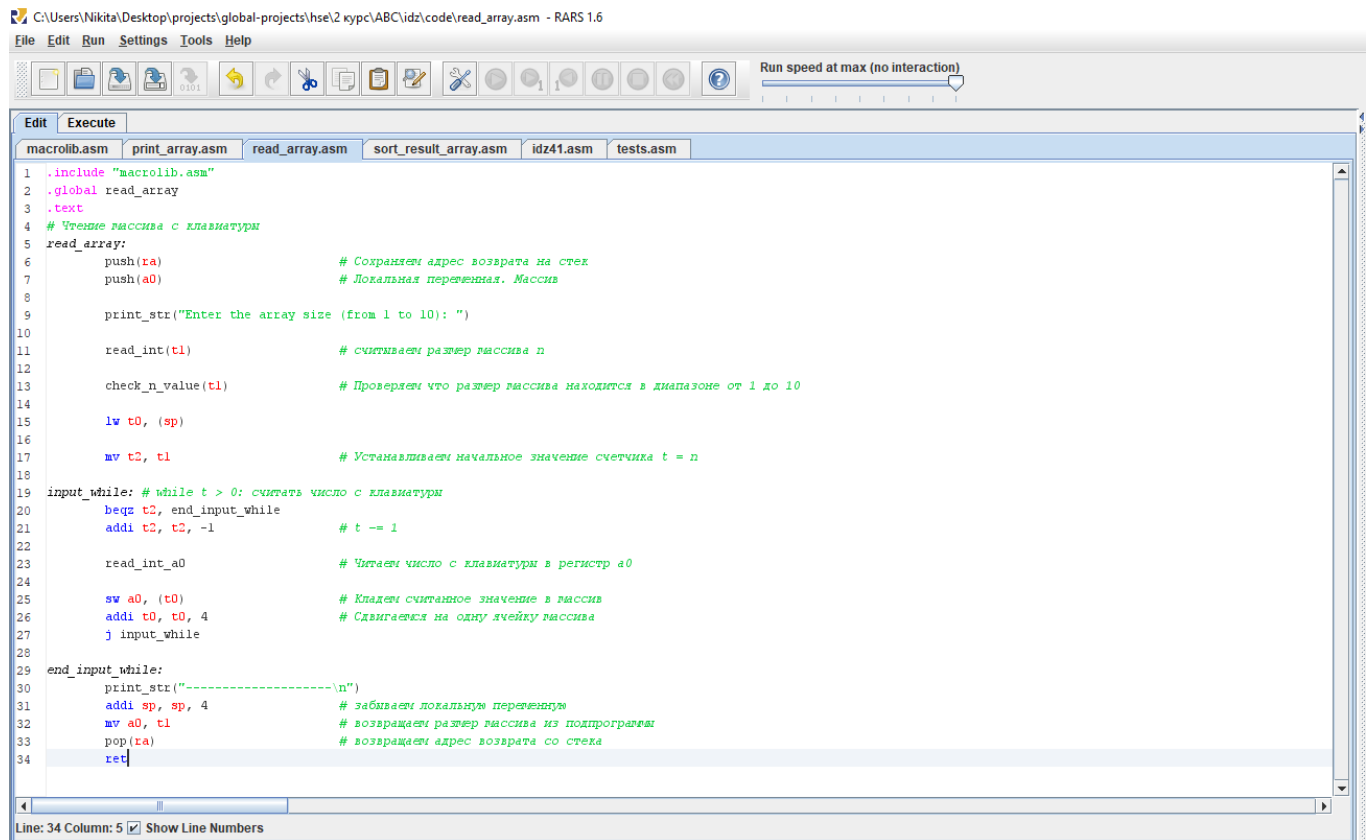
№ 3

Допускается использование требуемых подпрограмм без параметров и

локальных переменных.

Требование выполняется, поскольку подпрограммы есть и вынесены в отдельные модули.

Подпрограмма ввода данных с клавиатуры вынесенная в отдельный модуль:



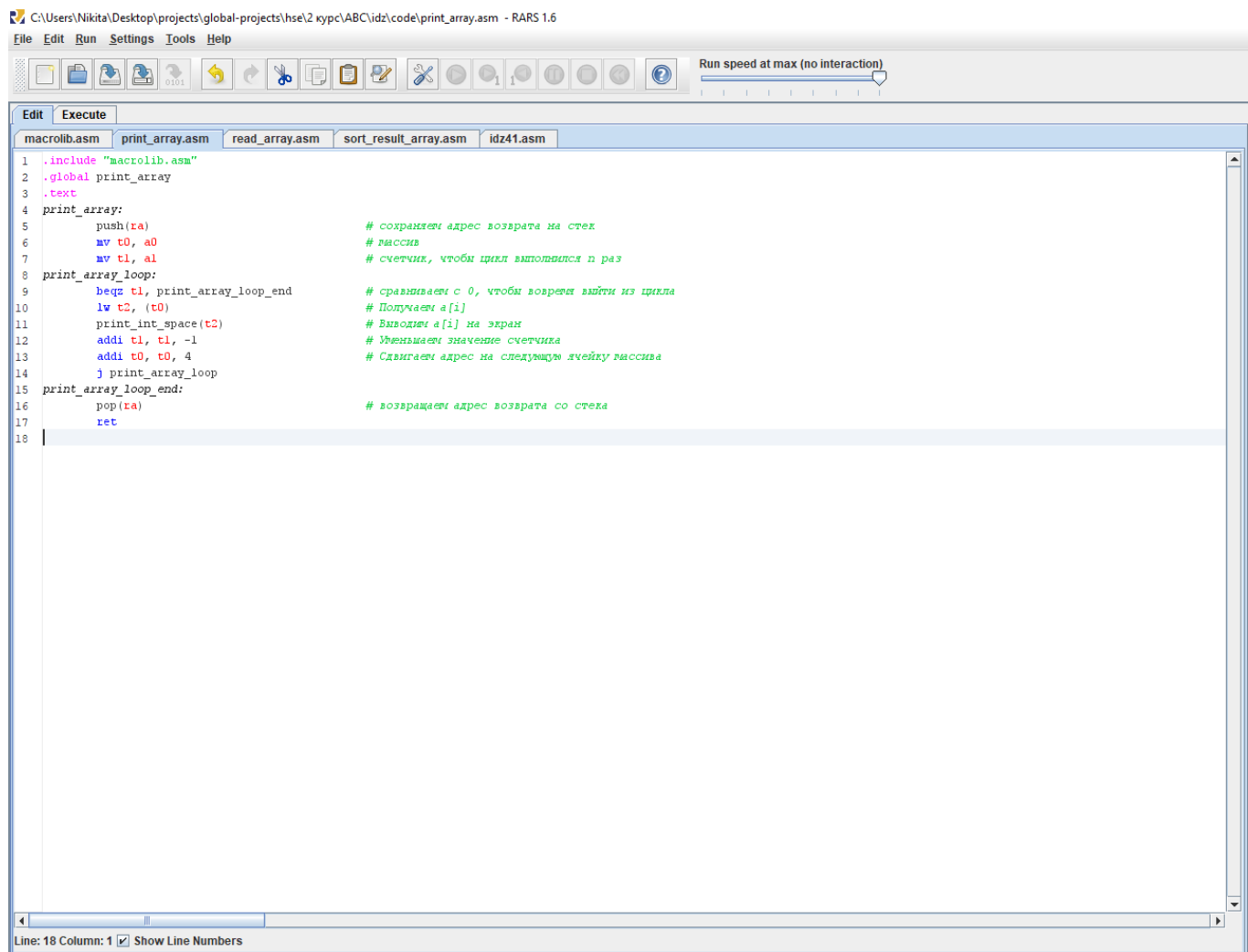
```
C:\Users\Nikita\Desktop\projects\global-projects\hse\2 кypc\ABC\idz\code\read_array.asm - RARS 1.6
File Edit Run Settings Tools Help
Run speed at max (no interaction)

macrolib.asm print_array.asm read_array.asm sort_result_array.asm idz41.asm tests.asm

1 .include "macrolib.asm"
2 .global read_array
3 .text
4 # Чтение массива с клавиатуры
5 read_array:
6     push(ra)                # Сохраняем адрес возврата на стек
7     push(a0)                # Локальная переменная. Массив
8
9     print_str("Enter the array size (from 1 to 10): ")
10
11     read_int(t1)            # считываем размер массива n
12
13     check_n_value(t1)       # Проверяем что размер массива находится в диапазоне от 1 до 10
14
15     lw t0, (sp)
16
17     mv t2, t1               # Устанавливаем начальное значение счетчика t = n
18
19 input_while: # while t > 0: считать число с клавиатуры
20     beqz t2, end_input_while
21     addi t2, t2, -1         # t -= 1
22
23     read_int_a0             # Читаем число с клавиатуры в регистр a0
24
25     sw a0, (t0)             # Кладем считанное значение в массив
26     addi t0, t0, 4          # Сдвигаемся на одну ячейку массива
27     j input_while
28
29 end_input_while:
30     print_str("-----\n")
31     addi sp, sp, 4          # забываем локальную переменную
32     mv a0, t1               # возвращаем размер массива из подпрограммы
33     pop(ra)                # возвращаем адрес возврата со стека
34     ret

Line: 34 Column: 5 Show Line Numbers
```

Подпрограммы вывода данных (вывода значений массива на экран) вынесенная в отдельный модуль:



Подпрограмма сортировки массива вынесенная в отдельный модуль:

C:\Users\Nikita\Desktop\projects\global-projects\hse\2 kypc\ABC\idz\code\sort_result_array.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

macrolib.asm print_array.asm read_array.asm sort_result_array.asm idz41.asm

```
1 .include "macrolib.asm"
2 .global sort_result_array
3 .text
4 sort_result_array:
5     push(ra)           # Сохраняем адрес возврата на стек
6     push(s1)           # Сохраняем значение из s1 на стек
7     push(s2)           # Сохраняем значение из s2 на стек
8     push(a0)           # Локальная переменная. В ней находится количество элементов в массиве
9     push(a1)           # Локальная переменная. В ней находится массив A
10    push(a2)           # Локальная переменная. В ней находится массив B
11
12    li s1, 0            # i
13 for_i_loop:
14    lw t0, 0(sp)
15    beq s1, t0, end_for_i_loop # Если i достиг n, то выходим из внешнего цикла
16    addi s2, s1, 1      # j = i + 1
17 for_j_loop:
18    lw t0, 0(sp)
19    beq s2, t0, end_for_j_loop # Если j достиг n, то выходим из внутреннего цикла
20    compare_i_and_j(s1, s2) # Сравнение i-го и j-го элементов массива. В a0 заносится
21    beqz a0, end_compare # Если i < j, то не меняем местами
22    swap_i_and_j(s1, s2) # Меняем местами i-ый и j-ый элементы массива
23 end_compare:
24    addi s2, s2, 1      # j += 1
25    j for_j_loop
26 end_for_j_loop:
27    addi s1, s1, 1      # i += 1
28    j for_i_loop
29 end_for_i_loop:
30    addi sp, sp, 12 # забываем три локальных переменных
31    pop(s2)          # Возвращаем значение s2 со стека в s2
32    pop(s1)          # Возвращаем значение s1 со стека в s1
33    pop(ra)          # Восстанавливаем адрес возврата со стека
34    ret
35
```

Line: 35 Column: 1 ☒ Show Line Numbers

№ 4

В отчете должно быть представлено полное тестовое покрытие. Приведены результаты тестовых прогонов. Например, с использованием скриншотов.



Edit Execute

macrolib.asm print_array.asm read_array.asm sort_result_array.asm idz41.asm tests.asm

```
1  # комментарии
2  # тесты
3
4  .include "macrolib.asm"
5  .global main
6  .data
7  .align 2
8  A_array: .space 40
9  A_arrend:
10 .align 2
11 B_array: .space 40
12 B_arrend:
13 .text
14 main:
15     read_array_wrapper(A_array) # читаем исходный массив с клавиатуры
16     mv s1, a0
17     init_array_by_indices(s1, B_array) # инициализируем массив B индексами
18     sort_result_array_wrapper(s1, A_array, B_array) # Сортируем массив B
19     print_array_wrapper(s1, B_array) # печатаем результат на экран
20     exit()
21
```

Line: 21 Column: 1 ☒ Show Line Numbers

Messages

Run I/O

```
Enter the array size (from 1 to 10): 6
4
2
3
7
1
5
-----
4 1 2 0 5 3
-- program is finished running (0) --

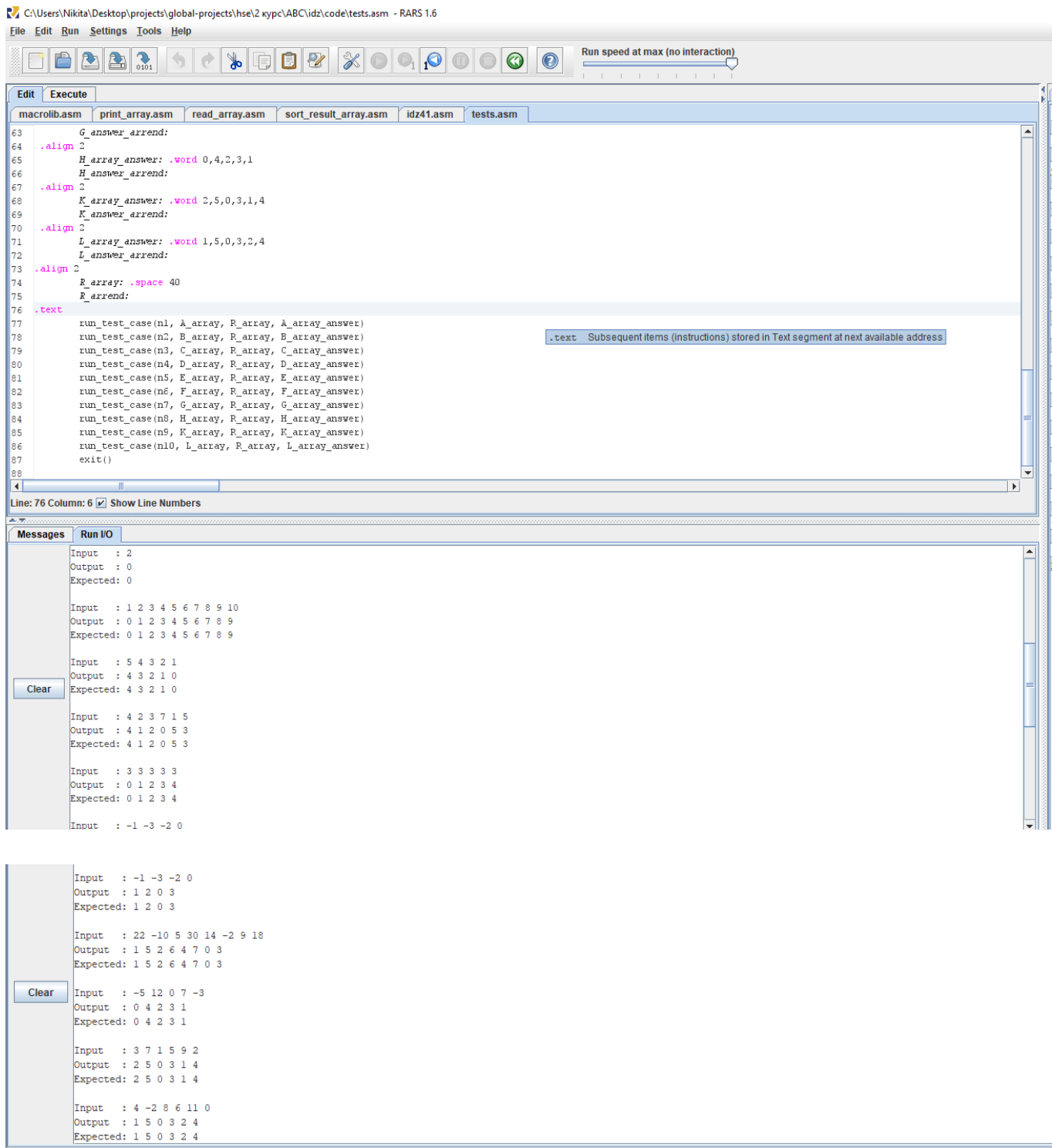
Enter the array size (from 1 to 10): -1
Invalid array size. Size must be between 1 and 10

-- program is finished running (0) --

Enter the array size (from 1 to 10): 11
Invalid array size. Size must be between 1 and 10

-- program is finished running (0) --
```

Clear



Требования на оценку 6-7:

№ 5

В программе необходимо использовать подпрограммы с передачей аргументов через соответствующие регистры, определяемые конвенцией по их использованию. При нехватке этих регистров для их хранения необходимо использовать стек.

Требование соблюдено.

Как видно из скриншотов пункта **№3** аргументы передаются подпрограммам в через регистры **a****

№ 6

Локальные переменные допускается размещать в свободных регистрах, при нехватке которых необходимо их сохранять на стеке. Или сразу все локальные переменные отображать на стек.

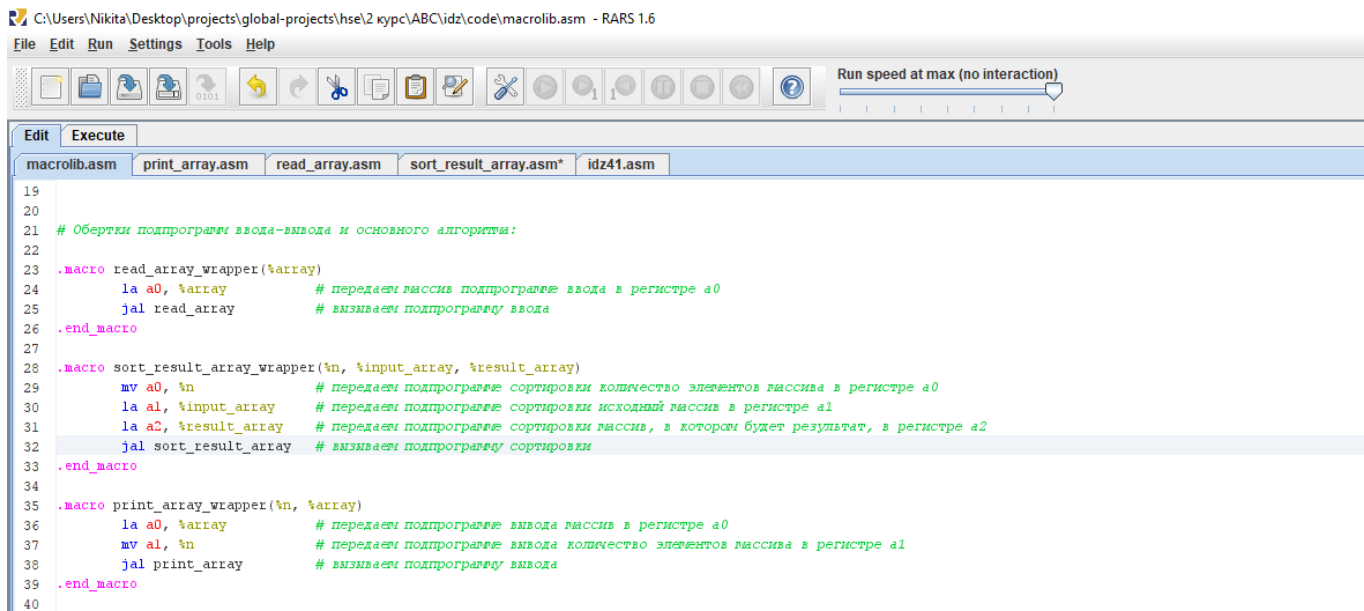
Требование соблюдено.

Как видно из скриншотов пункта **№3** локальные переменные находятся в регистрах **t**** или на стеке.

№ 7

В местах вызова функции добавить комментарии, описывающие передачу фактических параметров и перенос возвращаемого результата. При этом необходимо отметить, в каких регистрах или где на стеке отображаются соответствующие фактические параметры.

Требование выполняется, поскольку в месте вызова подпрограмм - в автономной библиотеке макроподстановок присутствуют комментарии по подготовке аргументов для их передачи в подпрограммы.



```
C:\Users\Nikita\Desktop\projects\global-projects\hse\2 курс\ABC\idz\code\macrolib.asm - RARS 1.6
File Edit Run Settings Tools Help
Run speed at max (no interaction)
macrolib.asm print_array.asm read_array.asm sort_result_array.asm* idz41.asm
19
20
21 # Обертки подпрограмм ввода-вывода и основного алгоритма:
22
23 .macro read_array_wrapper(%array)
24     la a0, %array          # передаем массив подпрограмме ввода в регистре a0
25     jal read_array         # вызываем подпрограмму ввода
26 .end_macro
27
28 .macro sort_result_array_wrapper(%n, %input_array, %result_array)
29     mv a0, %n              # передаем подпрограмме сортировки количество элементов массива в регистре a0
30     la a1, %input_array    # передаем подпрограмме сортировки исходный массив в регистре a1
31     la a2, %result_array   # передаем подпрограмме сортировки массив, в котором будет результат, в регистре a2
32     jal sort_result_array  # вызываем подпрограмму сортировки
33 .end_macro
34
35 .macro print_array_wrapper(%n, %array)
36     la a0, %array          # передаем подпрограмме вывода массив в регистре a0
37     mv a1, %n              # передаем подпрограмме вывода количество элементов массива в регистре a1
38     jal print_array       # вызываем подпрограмму вывода
39 .end_macro
40
```

№ 8

Информацию о проведенных изменениях отобразить в отчете наряду с информацией, необходимой на предыдущую оценку.

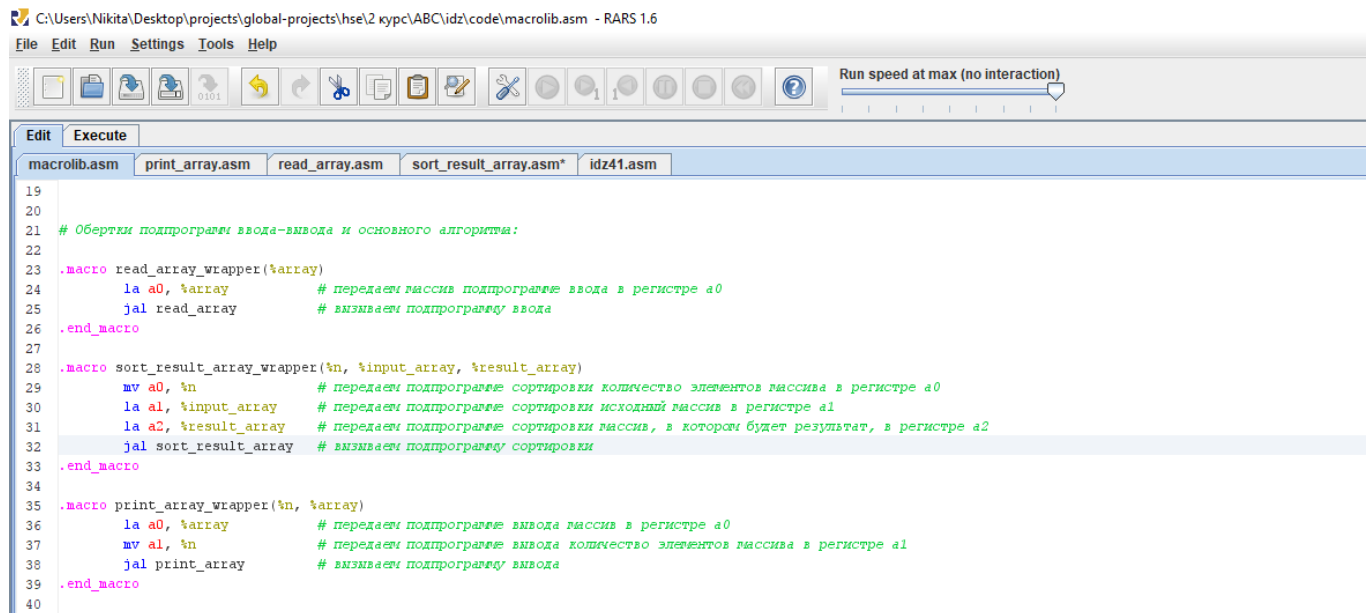
Вся необходимая информация приведена.

Требования на оценку 8:

№ 9

Разработанные подпрограммы должны поддерживать многократное использование с различными наборами исходных данных, включая возможность подключения различных исходных и результирующих массивов. То есть, поддерживать работу с формальными и фактическими параметрами.

Вызов подпрограмм и передача параметров им осуществляется внутри макроподстановок и их параметры не фиксированы, что позволяет использовать подпрограммы с различными наборами исходных данных.



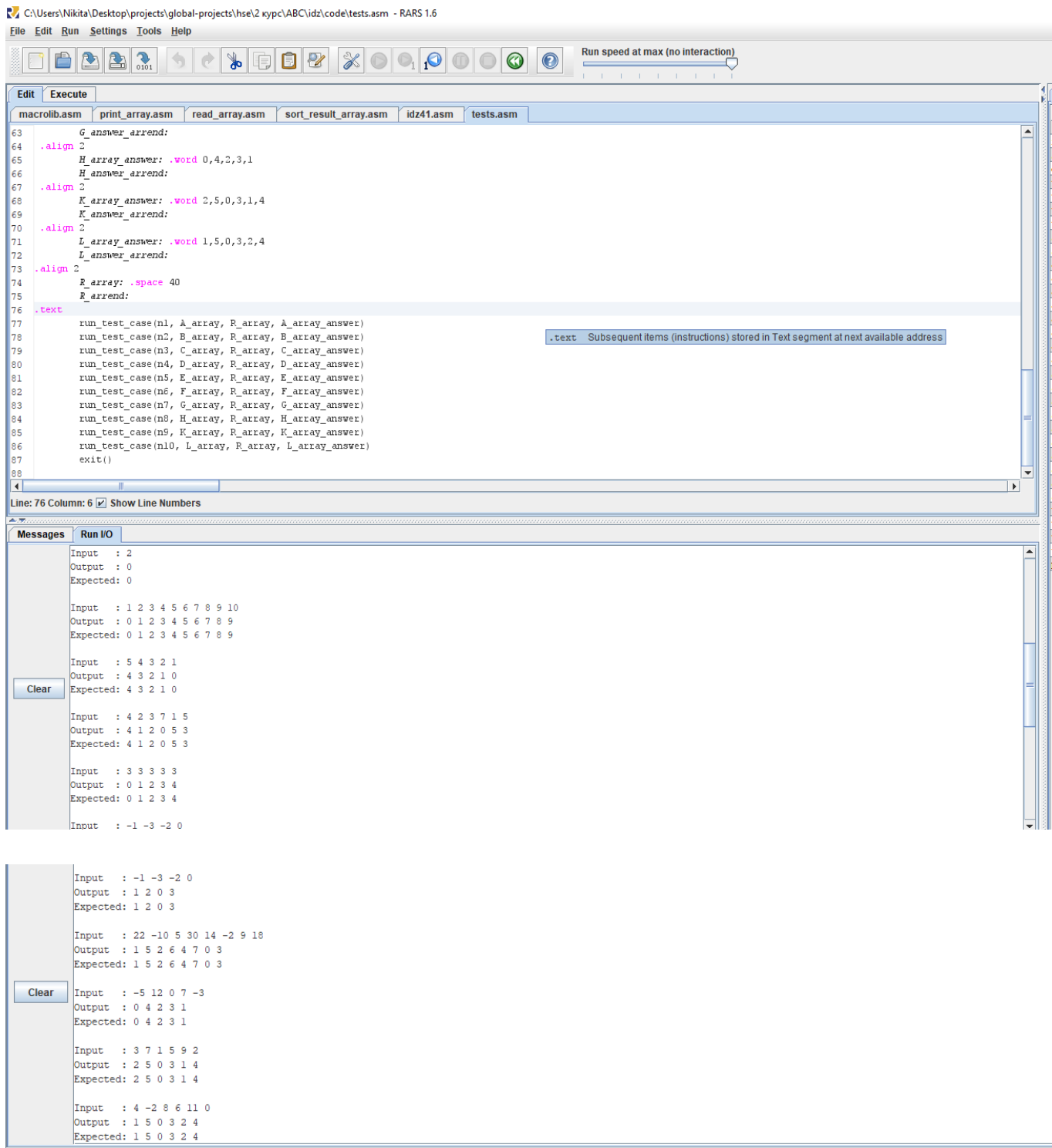
```
C:\Users\Nikita\Desktop\projects\global-projects\hse\2 курс\ABC\idz\code\macrolib.asm - RARS 1.6
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
macrolib.asm print_array.asm read_array.asm sort_result_array.asm* idz41.asm
19
20
21 # Обертки подпрограммы ввода-вывода и основного алгоритма:
22
23 .macro read_array_wrapper(%array)
24     la a0, %array      # передаем массив подпрограмме ввода в регистре a0
25     jal read_array      # вызываем подпрограмму ввода
26 .end_macro
27
28 .macro sort_result_array_wrapper(%input_array, %result_array)
29     mv a0, %input_array # передаем подпрограмме сортировки количество элементов массива в регистре a0
30     la a1, %input_array  # передаем подпрограмме сортировки исходный массив в регистре a1
31     la a2, %result_array # передаем подпрограмме сортировки массив, в котором будет результат, в регистре a2
32     jal sort_result_array # вызываем подпрограмму сортировки
33 .end_macro
34
35 .macro print_array_wrapper(%array)
36     la a0, %array      # передаем подпрограмме вывода массив в регистре a0
37     mv a1, %array      # передаем подпрограмме вывода количество элементов массива в регистре a1
38     jal print_array     # вызываем подпрограмму вывода
39 .end_macro
40
```

№ 10

Реализовать автоматизированное тестирование за счет создания дополнительной тестовой программы, осуществляющей прогон подпрограммы обработки массивов с различными тестовыми данными (вместо ввода данных). Осуществить прогон тестов обеспечивающих покрытие различных ситуаций. Тестовые данные можно формировать в различных исходных массивах

[Полный код программы осуществляющей автоматизированное тестирования.](#)

Приведу скриншоты работы этой программы:



№ 11

Добавить информацию о проведенных изменениях в отчет

Вся необходимая информация приведена.

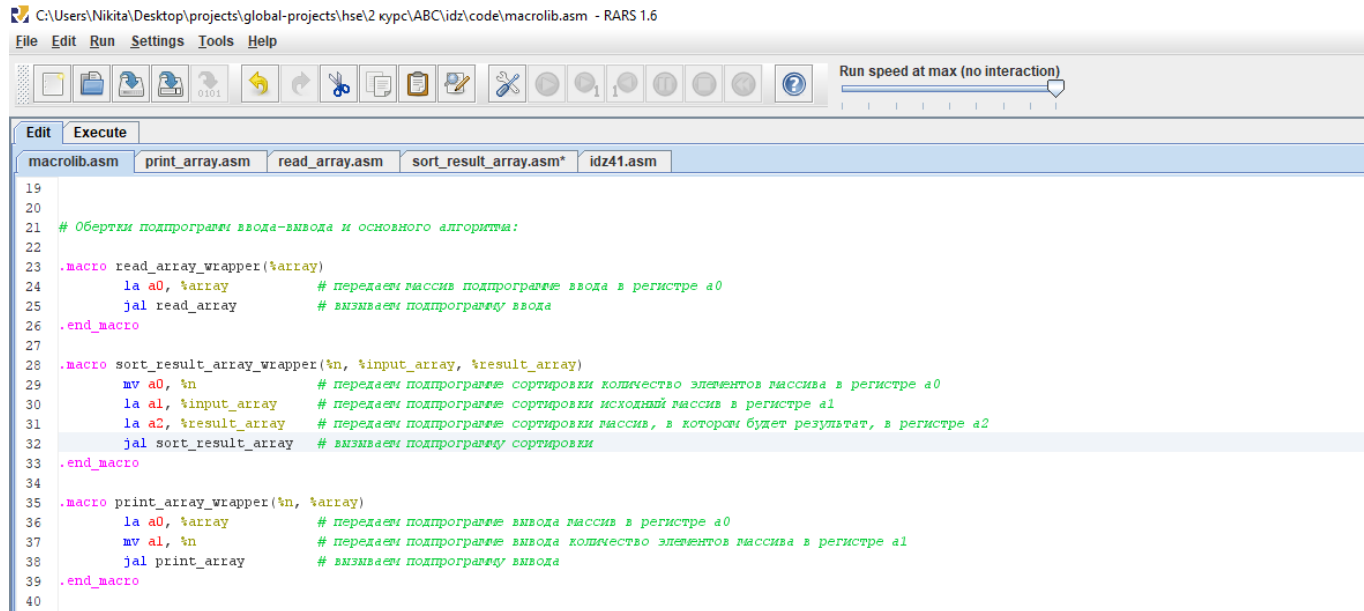
Требования на оценку 9:

№ 12

Добавить в программу использование макросов которые должны использоваться

в качестве оберток ранее написанных подпрограмм ввода и вывода данных, генерации тестовых массивов. Макросы должны поддерживать повторное использование с различными массивами и другими необходимыми параметрами

Требование выполняется, поскольку в автономной библиотеке макроподстановок присутствуют макро, которые подготавливают аргументы для подпрограмм и вызывают их внутри себя.



```
19
20
21 # Обертки подпрограмм ввода-вывода и основного алгоритма:
22
23 .macro read_array_wrapper(%array)
24     la a0, %array          # передаем массив подпрограмме ввода в регистре a0
25     jal read_array         # вызываем подпрограмму ввода
26 .end_macro
27
28 .macro sort_result_array_wrapper(%input_array, %result_array)
29     mv a0, %input_array     # передаем подпрограмме сортировки количество элементов массива в регистре a0
30     la a1, %input_array     # передаем подпрограмме сортировки исходный массив в регистре a1
31     la a2, %result_array    # передаем подпрограмме сортировки массив, в котором будет результат, в регистре a2
32     jal sort_result_array   # вызываем подпрограмму сортировки
33 .end_macro
34
35 .macro print_array_wrapper(%array)
36     la a0, %array          # передаем подпрограмме вывода массив в регистре a0
37     mv a1, %array          # передаем подпрограмме вывода количество элементов массива в регистре a1
38     jal print_array        # вызываем подпрограмму вывода
39 .end_macro
40
```

№ 13

Допускается реализация в виде собственных макросов и других фрагментов программы.

Требование соблюдается, поскольку в [библиотеке макроподстановок](#), есть и другие фрагменты программ.

Требования на оценку 10:

№ 14

Программа должна быть разбита на несколько единиц компиляции (ассемблерных файлов). При этом подпрограммы ввода–вывода должны составлять унифицированные модули, используемые повторно как в программе, осуществляющей ввод или генерацию исходных данных, так и в программе, осуществляющей тестовое покрытие.

Требование выполняется, поскольку в проекте указанные подпрограммы находятся в унифицированных модулях.

№ 15

Макросы должны быть выделены в отдельную автономную библиотеку

Требование выполняется, поскольку все макроподстановки вынесены в отдельную библиотеку.

[Исходный код проекта](#)