

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 ==изначальная сумма + новый рассматриваемый элемент; в завершении, среднее арифметическое = сумма / 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:

C:\HSE\Assembler\Горбачева_Ассемблер_ИД31\main.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Floating Point Control and Status

Name	Number	Value
ustatus	0	0x00000000
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x00000000
useratch	64	0x00000000
uepc	65	0x00000000
ucause	66	0x00000000
utval	67	0x00000000
uip	68	0x00000000
cycle	3072	0x00000000
time	3073	0x00000000
instret	3074	0x00000000
cycleh	3200	0x00000000
timelh	3201	0x00000000
instrth	3202	0x00000000

Registers Floating Point Control and Status

```

1 #main.asm - основная программа с меню и автотестами
2
3 .include "io.asm"      #модуль ввода-вывода
4 .include "math.asm"    #модуль математических вычислений
5 .include "file_io.asm" #модуль работы с файлами
6
7 .data #текущие значения
8 n: .word 0           #размер массива
9 array_A: .space 40   #массив A (10 элементов * 4 байта) согласно условию
10 array_B: .space 40   #массив B (10 элементов * 4 байта) согласно условию
11
12 prompt_start: .asciz "\nMENU\nEnter '0': input your array.\nEnter other symbols: launch autotests.\n\nYour choice: "
13 prompt_next: .asciz "If you want to finish program, enter '0', otherwise press any other number to restart: "
14 sep: .asciz " "
15 newline: .asciz "\n"
16
17 msg_array_A: .asciz "Массив A: "
18 msg_array_B: .asciz "Массив B: "
19 msg_avg: .asciz "Среднее арифметическое: "
20 msg_error: .asciz "Символ Некорректный размер массива.\n"
21 msg_input_choice: .asciz "Выберите способ ввода:\n1 - Клавиатура\n2 - Файл\nВедите выбор: "
22 msg_invalid_input: .asciz "Неверный выбор ввода!\n"
23
24
25 .text
26 .global main
27
28 main:
29     # вывод меню выбора
30     li $t7, 4 #код системного вызова для печати строки
31     la $t0, prompt_start #адрес строки с меню в регистр 40
32     call #вызываем систему для печати строки
33
34     # Чтение числа

```

Line: 115 Column: 63 Show Line Numbers

Messages Run I/O

Clear ✓ Тест пройден успешно

C:\HSE\Assembler\Горбачева_Ассемблер_ИД31\file.io.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Floating Point Control and Status

Name	Number	Value
ustatus	0	0x00000000
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x00000000
useratch	64	0x00000000
uepc	65	0x00000000
ucause	66	0x00000000
utval	67	0x00000000
uip	68	0x00000000
cycle	3072	0x00000000
time	3073	0x00000000
instret	3074	0x00000000
cycleh	3200	0x00000000
timelh	3201	0x00000000
instrth	3202	0x00000000

Registers Floating Point Control and Status

```

1 # file.io.asm - модуль для работы с файлами
2
3 .data
4 msg_filename: .asciz "Введите имя файла с данными: "
5 msg_success: .asciz "Файл успешно прочитан. Размер: "
6 msg_file_error: .asciz "Символ файла не существует или поврежден!\n"
7 msg_reading: .asciz "Чтение данных из файла...\n"
8 file_newline: .asciz "\n"
9 filename: .space 64 # Буфер для имени файла (64 байта)
10 file_buffer: .space 256 # Буфер для чтения файла (256 байт)
11
12 .text
13 # Подпрограмма чтения массива из файла
14 # Параметры: a0 - адрес массива для загрузки
15 # Возвращают: a0 - размер массива, a1 - 0 при успехе, -1 при ошибке
16
17 read_from_file:
18     addi $t0, $t0, -16 # Выделяем 16 байт в стеке (4 слова) для сохранения регистров
19     sw $t0, 12($sp) # Сохраняем адрес возврата (return address) по смещению 12
20     sw $t0, 8($sp) # Сохраняем a0 (адрес целевого массива) по смещению 8
21     sw $t1, 4($sp) # Сохраняем a1 (дескриптор файла) по смещению 4
22     sw $t2, 0($sp) # Сохраняем a2 (временное хранение) по смещению 0
23
24     mv $t0, a0 # Компилируем адрес целевого массива из a0 в a0 (сохраняем параметр)
25
26     # Ввод имени файла с клавиатуры
27     la $t0, msg_filename # Загружаем адрес сообщения "Введите имя файла..." в a0
28     li $t7, 4 # Загружаем код системного вызова для печати строки (4)
29     call #Вызываем систему - печатаем приглашение
30
31     la $t0, filename # Загружаем адрес буфера для имени файла в a0
32     li $t1, 64 # Указываем максимальную длину вводимой строки (64 символа)
33     li $t2, 8 # Загружаем код системного вызова для ввода строки (8)

```

Line: 228 Column: 62 Show Line Numbers

Messages Run I/O

Clear ✓ Тест пройден успешно

C:\HSE\assembler\Горбачева_Ассемблер_ИД31\io.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Registers Floating Point Control and Status

Name	Number	Value
ustatus	0	0x00000000
fflags	1	0x00000000
fmr	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x00000000
usecratch	64	0x00000000
uepc	65	0x00000000
ucause	66	0x00000000
utval	67	0x00000000
uip	68	0x00000000
cycle	3072	0x00000000
time	3073	0x00000000
instrct	3074	0x00000000
cycleh	3200	0x00000000
timeth	3201	0x00000000
instrth	3202	0x00000000

```

1 # io.asm - модуль ввода-вывода для работы с консолью
2 # Содержит подпрограммы для ввода и вывода массивов
3
4 .data
5 # Сообщения и форматы для ввода-вывода
6 msg_element: .asciz "A"      # Начало приглашения для ввода элемента
7 msg_bracket: .asciz "]"      # Окончание приглашения
8 msg_comma: .asciz ","        # Разделитель элементов при выводе
9 io_newline: .asciz "\n"       # Символ новой строки
10
11 .text
12
13 # Подпрограмма ввода целого числа с консоли
14 # Параметры: a0 - адрес строки-приглашения для ввода
15 # Возвращает: a0 - введенное целое число
16 input_int:
17     addi sp, sp, -4          # Выделение места в стеке
18     sw r4, 0(sp)            # Сохранение адреса возврата
19
20     li r7, 4                # Вывод строки-приглашения
21     ecall #выходная строка-приглашение
22     li r7, 5                # Ввод целого числа
23     ecall #Результат в a0
24
25     lw r4, 0(sp)            # восстановление регистра
26     addi sp, sp, 4           # восстановление места в стеке
27     ret
28
29 # Подпрограмма ввода массива с консоли
30 # Параметры: a0 - адрес массива, a1 - размер массива
31
32 input_array:
33     addi sp, sp, -16         #сохранение регистра
4 Line: 122 Column: 1 Show Line Numbers

```

Messages Run I/O

Clear ✓ Тест пройден успешно

C:\HSE\assembler\Горбачева_Ассемблер_ИД31\math.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Registers Floating Point Control and Status

Name	Number	Value
ustatus	0	0x00000000
fflags	1	0x00000000
fmr	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x00000000
usecratch	64	0x00000000
uepc	65	0x00000000
ucause	66	0x00000000
utval	67	0x00000000
uip	68	0x00000000
cycle	3072	0x00000000
time	3073	0x00000000
instrct	3074	0x00000000
cycleh	3200	0x00000000
timeth	3201	0x00000000
instrth	3202	0x00000000

```

1 # math.asm - модуль математических вычислений для работы с массивами
2 # Содержит подпрограммы для вычисления среднего и формирования массива B
3
4 .text
5
6
7 # Подпрограмма вычисления среднего арифметического массива
8 # Параметры: a0 - адрес массива, a1 - размер массива
9 # Возвращает: a0 - среднее арифметическое (число с плавающей точкой)
10 calculate_avg:
11     addi sp, sp, -20         #сохранение регистра
12     sw r4, 16(sp)
13     sw r0, 12(sp)           # a0 - адрес массива
14     sw r1, 8(sp)             # a1 - размер массива
15     sw r2, 4(sp)             # a2 - счетчик i
16     sw r3, 0(sp)             # a3 - сумма элементов
17
18     mv r0, a0                # Сохранение параметров
19     mv r1, a1
20     li r2, 0                  # Инициализация счетчика i = 0
21     li r3, 0                  # Инициализация суммы sum = 0
22
23 sum_loop:
24     bge r2, r1, sum_end # Условие выхода: i >= N
25
26     lw r0, 0(r0)              # Загрузка array[i]
27     add r3, r3, r0            # sum = sum + array[i]
28
29     addi r0, r0, 4             # Следующий элемент массива
30     addi r2, r2, 1             # Увеличение счетчика
31     j sum_loop               # Повторение цикла
32
33 4 Line: 1 Column: 16 Show Line Numbers

```

Messages Run I/O

Clear ✓ Тест пройден успешно

The screenshot shows the RARS 1.6 assembly editor interface. The assembly code in the main window is as follows:

```

1 # file_io.asm - модуль для работы с файлами
2
3 .data
4 msg_filename: .asciz "Введите имя файла с данными: "
5 msg_success: .asciz "Файл успешно прочитан. Размер: "
6 msg_file_error: .asciz "Символы файла не существуют или повреждены!\n"
7 msg_reading: .asciz "Чтение данных из файла...\n"
8 file_newline: .asciz "\n"
9 filename: .space 64 # Буфер для имени файла (64 байта)
10 file_buffer: .space 256 # Буфер для чтения файла (256 байт)
11
12 .text
13 # Подпрограмма чтения массива из файла
14 # Переметры: a0 - адрес массива для заполнения
15 # Возвращает: a0 - размер массива, a1 - 0 при успехе, -1 при ошибке
16
17 read_from_file:
18 addi sp, sp, -16 # Выделаем 16 байт в стеке (4 слова) для сохранения регистров
19 mtc r4, 12(sp) # Сохраняем адрес возврата (return address) по смещению 12
20 mtc a0, 8(sp) # Сохраняем a0 (адрес целевого массива) по смещению 8
21 mtc s1, 4(sp) # Сохраняем s1 (дескриптор файла) по смещению 4
22 mtc s2, 0(sp) # Сохраняем s2 (временное хранение) по смещению 0
23
24 mv a0, a0 # Конструируем адрес целевого массива из a0 и s0 (сохранением параметра)
25
26 # Ввод имени файла с клавиатуры
27 la a0, msg_filename # Загружаем адрес сообщения "Введите имя файла..." в a0

```

The Registers window shows the following register values:

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7ffffefc
gp	3	0x10000000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
t10	26	0x00000000
t11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400f94

The Messages window shows the following output:

```

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

Your choice:

```

Выполненные условия:

На 4-5:

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

На 6-7:

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

На 8:

- Многократное использование подпрограмм.
- Реализована дополнительная тестовая программа с автоматическими тестами (tests.asm).
- 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.