

# Отчет по ИДЗ\_1

Подготовил Растворов Сергей Евгеньевич БПИ236

## Вариант 18

### Правило формирования массива В:

Сформировать массив В из элементов массива А уменьшением всех элементов до первого положительного на 5.

### Тесты ручные

```
Enter the number of elements: 3
Enter element: 1
Enter element: 2
Enter element: 3
Array B: 1 2 3

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

```
Enter the number of elements: 3
Enter element: -1
Enter element: -2
Enter element: -3
Array B: -6 -7 -8

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

```
Enter the number of elements: 5
Enter element: -2
Enter element: -33
Enter element: -23
Enter element: 3
Enter element: 22
Array B: -7 -38 -28 3 22

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

```
Enter the number of elements: 1000
Error: invalid number of elements!
Enter the number of elements: -2
Error: invalid number of elements!
```

```
Enter the number of elements: 0
Error: invalid number of elements!
Enter the number of elements: 11
Error: invalid number of elements!
Enter the number of elements: --
err
```

Тесты автоматические имеются внутри программы

## Результаты

Написан код, оформлены тесты.

Пояснительная записка:

Прошу прощения за просроченную сдачу работы.

Я участвовал в вышкинском хакатоне и поздно занялся решением ИДЗ.

Буду очень рад, если мое решение будет рассмотрено.

## О работе

Написан код для ввода массива A и формирования массива B на основе заданного правила.

Реализованы подпрограммы для ввода и вывода массивов с передачей параметров через регистры.

### Тестирование программы:

Создана отдельная подпрограмма для автоматического тестирования, которая выполняет тесты на различных наборах данных и выводит результаты.

В процессе тестирования проверялась корректность работы программы с крайними и обычными значениями.

### Организация структуры проекта:

Проект разбит на несколько файлов для улучшения читаемости и повторного использования:

main.asm: основной файл с логикой программы.

data.asm: файл с определениями сообщений и массивов.

macros.asm: библиотека с макросами для ввода и вывода.

### Реализация пользовательского интерфейса:

Добавлено меню выбора режима работы (ручной ввод или автоматическое тестирование).

Все сообщения представлены на английском языке для обеспечения переносимости кода.

## Код программы

```

.include "macros.asm"
.include "data.asm"
.text
.globl main

main:
    li t5, 1
    li t6, 2
    print_string msg_select_mode
    print_string msg_ur_ans
    input_int a0
    beq a0, t5, manual_mode
    beq a0, t6, test_automatic
    print_string msg_newline
    j main

manual_mode:
    input_size t0
    input_array array_a, t0
    form_array array_a, array_b, t0
    print_string msg_output_b
    output_array array_b, t0
    j end

end:
    li a7, 10
    ecall

test_automatic:
    li a0, 3
    jal test_case_0
    li a0, 3
    jal test_case_1
    li a0, 5
    jal test_case_2
    li a0, 0
    jal test_case_3
    j end

# Тест 0: {1, 2, 3}
test_case_0:
    li t0, 3
    la t1, array_a
    li t2, 1
    sw t2, 0(t1)
    li t2, 2
    sw t2, 4(t1)
    li t2, 3
    sw t2, 8(t1)
    # Загрузка адреса array_a

```

```
print_string msg_output_a
output_array array_a, t0          # Вывод массива A
form_array array_a, array_b, t0
print_string msg_output_b
output_array array_b, t0          # Вывод массива B
j end_test
```

# Тест 1: {-1, -2, -3}

```
test_case_1:
    li t0, 3
    la t1, array_a
    li t2, -1
    sw t2, 0(t1)
    li t2, -2
    sw t2, 4(t1)
    li t2, -3
    sw t2, 8(t1)
    print_string msg_output_a
    output_array array_a, t0
    form_array array_a, array_b, t0
    print_string msg_output_b
    output_array array_b, t0
    j end_test
```

# Тест 2: {-2, -33, -23, 3, 22}

```
test_case_2:
    li t0, 5
    la t1, array_a
    li t2, -2
    sw t2, 0(t1)
    li t2, -33
    sw t2, 4(t1)
    li t2, -23
    sw t2, 8(t1)
    li t2, 3
    sw t2, 12(t1)
    li t2, 22
    sw t2, 16(t1)
    print_string msg_output_a
    output_array array_a, t0
    form_array array_a, array_b, t0
    print_string msg_output_b
    output_array array_b, t0
    j end_test
```

# Тест 3: Проверка на некорректный ввод

```
test_case_3:
    print_string msg_prompt_n
    mv t0, a0
```

```

        output_int t0
        print_string msg_space
        print_string msg_error
        j end_test

end_test:
        print_string msg_newline
        ret

```

```

.data
msg_prompt_n:      .asciz "Enter the number of elements: "
msg_prompt_elem:   .asciz "Enter element: "
msg_output_a:      .asciz "Array A: "
msg_output_b:      .asciz "Array B: "
msg_newline:       .asciz "\n"
msg_space:         .asciz " "
msg_error:         .asciz "Error: invalid number of elements!\n"
msg_select_mode:   .asciz "Select mode:\n1. Manual input\n2.
Automatic test\n"
msg_ur_ans:        .asciz "_: "

max_size:          .word 10
array_a:           .space 40  # 10 элементов по 4 байта (целое число)
array_b:           .space 40

```

# Макрос для чтения целого числа

```

.macro input_int %reg
        li a7, 5
        ecall
        mv %reg, a0
.end_macro

```

# Макрос для вывода строки

```

.macro print_string %str
        la a0, %str
        li a7, 4
        ecall
.end_macro

```

# Макрос для ввода размера массива с проверкой

```

.macro input_size %size_reg
        input_again:
        print_string msg_prompt_n
        input_int %size_reg
        li t1, 1
        li t2, 10

```

```

        blt %size_reg, t1, size_error
        bgt %size_reg, t2, size_error
        j input_size_done

size_error:
    print_string msg_error
    j input_again

input_size_done:
.end_macro

# Макрос для ввода массива
.macro input_array %addr, %size
    la t1, %addr
    mv t2, %size
    li t3, 0

input_array_loop:
    bge t3, t2, input_array_done
    print_string msg_prompt_elem
    li a7, 5
    ecall
    sw a0, 0(t1)
    addi t1, t1, 4
    addi t3, t3, 1
    j input_array_loop

input_array_done:
.end_macro

# Макрос для формирования массива B на основе массива A
.macro form_array %src_addr, %dest_addr, %size
    la t1, %src_addr
    la t2, %dest_addr
    mv t3, %size
    li t5, 0
    li t6, 1

form_loop:
    beqz t3, form_done
    lw t4, 0(t1)
    bgtz t4, copy_rest
    beq t5, t6, copy_elem
    addi t4, t4, -5

copy_elem:
    sw t4, 0(t2)
    addi t1, t1, 4
    addi t2, t2, 4

```

```

        addi t3, t3, -1
        j form_loop

copy_rest:
        li t5, 1
        j copy_elem

form_done:
.end_macro

# Макрос для вывода массива
.macro output_array %addr, %size
        la t1, %addr
        mv t2, %size
        li t3, 0

output_array_loop:
        bge t3, t2, output_array_done
        lw a0, 0(t1)
        li a7, 1
        ecall
        print_string msg_space
        addi t1, t1, 4
        addi t3, t3, 1
        j output_array_loop

output_array_done:
        print_string msg_newline
.end_macro

```