

ГУАП

КАФЕДРА № 14

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

старший преподаватель  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

Н. И. Синёв  
\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

ВЕТВЛЕНИЯ В АССЕМБЛЕРЕ

по курсу:

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ АССЕМБЛЕРА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

1245



24.03.24

\_\_\_\_\_  
подпись, дата

Г. С. Куранов  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2024

## Постановка задачи

Вариант: 9

Если сумма целых чисел A, B и C меньше единицы, то наименьшее из этих трех чисел заменить суммой двух других.

## Исходный код

```
.global _main  
.align 2  
_main:
```

//Если сумма целых чисел A, B и C меньше единицы, то наименьшее из этих трех чисел заменить суммой двух других.

```
mov x0, #-6 //ввод A  
mov x1, #2 //ввод B  
mov x2, #4 //ввод C
```

```
mov x3, #0
```

```
add x3, x0, x1 // A + B  
add x3, x3, x2 // A + B + C
```

```
cmp x3, #1  
blt delta  
b skip
```

delta:

```
cmp x0, x1  
b.le to_x0_x2  
b to_x1_x2
```

to\_x0\_x2:

```
cmp x0, x2  
b.le _x0  
b _x2
```

to\_x1\_x2:

```
cmp x1, x2  
b.lt _x1  
b _x2
```

\_x0:

```
add x0, x1, x2  
b skip
```

\_x1:

```
add x1, x0, x2  
b skip
```

\_x2:

```
add x2, x0, x1  
b skip
```

skip:

```
mov x3, x0
mov x4, x1
mov x5, x2
```

```
str x3, [sp] //сохраняем (store) число в стековый регистр
str x4, [sp, #8] //сохраняем (store) число в стековый регистр + 8 байт смещение
str x5, [sp, #16] //сохраняем (store) число в стековый регистр + 16 байт смещение
adr x0, output_str //загружаем адрес строки в x0
bl _printf //вызываем C-функцию вывода
```

```
mov x0, #0
mov X16, #1 //системный вызов 1 завершает программу
svc #0x80 //вызываем системную функцию с номером 1
```

output\_str:

```
.asciz "Output value: %d, %d, %d\n"
```

### Тестирование

1) В ручную:

input			output		
A	B	C	A'	B'	C'
-6	2	4	6	2	4
2	-3	2	2	-3	2
-298	400	-100	-298	400	-100

Таблица 1 – Результаты подсчетов

Подсчеты:

1. A = -6, B = 2, C = 4

Sum(A, B, C) = -6 + 2 + 4 = 0, 0 < 1 => min(A, B, C) = -6, -6 -> B + C = 2 + 4 = 6

2. A = 2, B = -3, C = 2

Sum(A, B, C) = 2 + -3 + 2 = 1, 1 !< 1 => A = A = 2, B = B = -3, C = C = 2

3. A = -298, B = 400, C = -100

Sum(A, B, C) = -298 + 400 - 100 = 2, 2 !< 1 => A = A = -298, B = B = 400,

C = C = -100

2) С помощью программы:

```
Assembler > lab_3_if > No Selection
1  .global _main
2  .align 2
3  _main:
4
5      //Если сумма целых чисел A, B и C меньше единицы, то наименьшее из этих трех чисел заменить суммой двух других.
6      mov x0, #-6 //ввод A
7      mov x1, #2 //ввод B
8      mov x2, #4 //ввод C
9
10     mov x3, #0
11
12     add x3, x0, x1 // A + B
13     add x3, x3, x2 // A + B + C
14
15     cmp x3, #1
16     blt delta
17     b skip
18
19 delta:
20     cmp x0, x1
21     b.le to_x0_x2
22     b to_x1_x2
23
24 to_x0_x2:
25     cmp x0, x2
26     b.le _x0
27     b _x2
28
29 to_x1_x2:
30     cmp x1, x2
31     b.lt _x1
32     b _x2
33
34 _x0:
35     add x0, x1, x2
36     b skip
37
38 _x1:
39     add x1, x0, x2
40     b skip
41
42 _x2:
```

Line: 62 Col: 40

Output value: 6, 2, 4  
Program ended with exit code: 0

Рисунок 1 - Результат работы программы со значениями A, B, C = -6, 2, 4

```
Assembler > lab_3_if > No Selection
1  .global _main
2  .align 2
3  _main:
4
5      //Если сумма целых чисел A, B и C меньше единицы, то наименьшее из этих трех чисел заменить суммой двух других.
6      mov x0, #2 //ввод A
7      mov x1, #-3 //ввод B
8      mov x2, #2 //ввод C
9
10     mov x3, #0
11
12     add x3, x0, x1 // A + B
13     add x3, x3, x2 // A + B + C
14
15     cmp x3, #1
16     blt delta
17     b skip
18
19 delta:
20     cmp x0, x1
21     b.le to_x0_x2
22     b to_x1_x2
23
24 to_x0_x2:
25     cmp x0, x2
26     b.le _x0
27     b _x2
28
29 to_x1_x2:
30     cmp x1, x2
31     b.lt _x1
32     b _x2
33
34 _x0:
35     add x0, x1, x2
36     b skip
37
38 _x1:
39     add x1, x0, x2
40     b skip
41
42 _x2:
```

Line: 8 Col: 15

Output value: 2, -3, 2  
Program ended with exit code: 0

Рисунок 2 - Результат работы программы со значениями A, B, C = 2, -3, 2

```
Assembler > lab_3_if > No Selection
1  .global _main
2  .align 2
3  _main:
4
5  //Если сумма целых чисел A, B и C меньше единицы, то наименьшее из этих трех чисел заменить суммой двух других.
6  mov x0, #-298 //ввод A
7  mov x1, #400 //ввод B
8  mov x2, #-100 //ввод C
9
10  mov x3, #0
11
12  add x3, x0, x1 // A + B
13  add x3, x3, x2 // A + B + C
14
15  cmp x3, #1
16  blt delta
17  b skip
18
19  delta:
20  cmp x0, x1
21  b.le to_x0_x2
22  b to_x1_x2
23
24  to_x0_x2:
25  cmp x0, x2
26  b.le _x0
27  b _x2
28
29  to_x1_x2:
30  cmp x1, x2
31  b.lt _x1
32  b _x2
33
34  _x0:
35  add x0, x1, x2
36  b skip
37
38  _x1:
39  add x1, x0, x2
40  b skip
41
42  _x2:
```

Line: 6 Col: 18

Output value: -298, 400, -100  
Program ended with exit code: 0

Рисунок 3 - Результат работы программы со значениями A, B, C = -298, 400, -100

### Выводы

В результате данной лабораторной работы мне удалось разработать программу на языке программирования Assembler, которая с помощью ветвлений определяла, является сумма меньше 1 или нет. Если сумма чисел A, B, C была меньше 1, то также с помощью ветвлений находилась минимальный элемент из этих трех и заменялся на сумму двух других. Также были проверены все 3 теста, которые были решены вручную. Результаты работы программы совпадают с просчитанными вручную, поэтому лабораторная работа выполнена успешно, цели ЛР достигнуты.