МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

кафедра программного обеспечения и администрирования

информационных систем

Отчёт

по лабораторной работе №2.1

«Арифметические операции с многобайтовыми данными»

по дисциплине

###### «Архитектура вычислительных систем»

|  |  |
| --- | --- |
| Выполнил: | студент группы 313  Козявин Максим Сергеевич |
| Проверил: | к.т.н., доцент  кафедры ПОиАИС  Жмакин А.П. |

Курск

2023

**Вариант задания:**



Рисунок 1 – Вариант задания

**Граф-схема алгоритма основной программы и подпрограмм:**

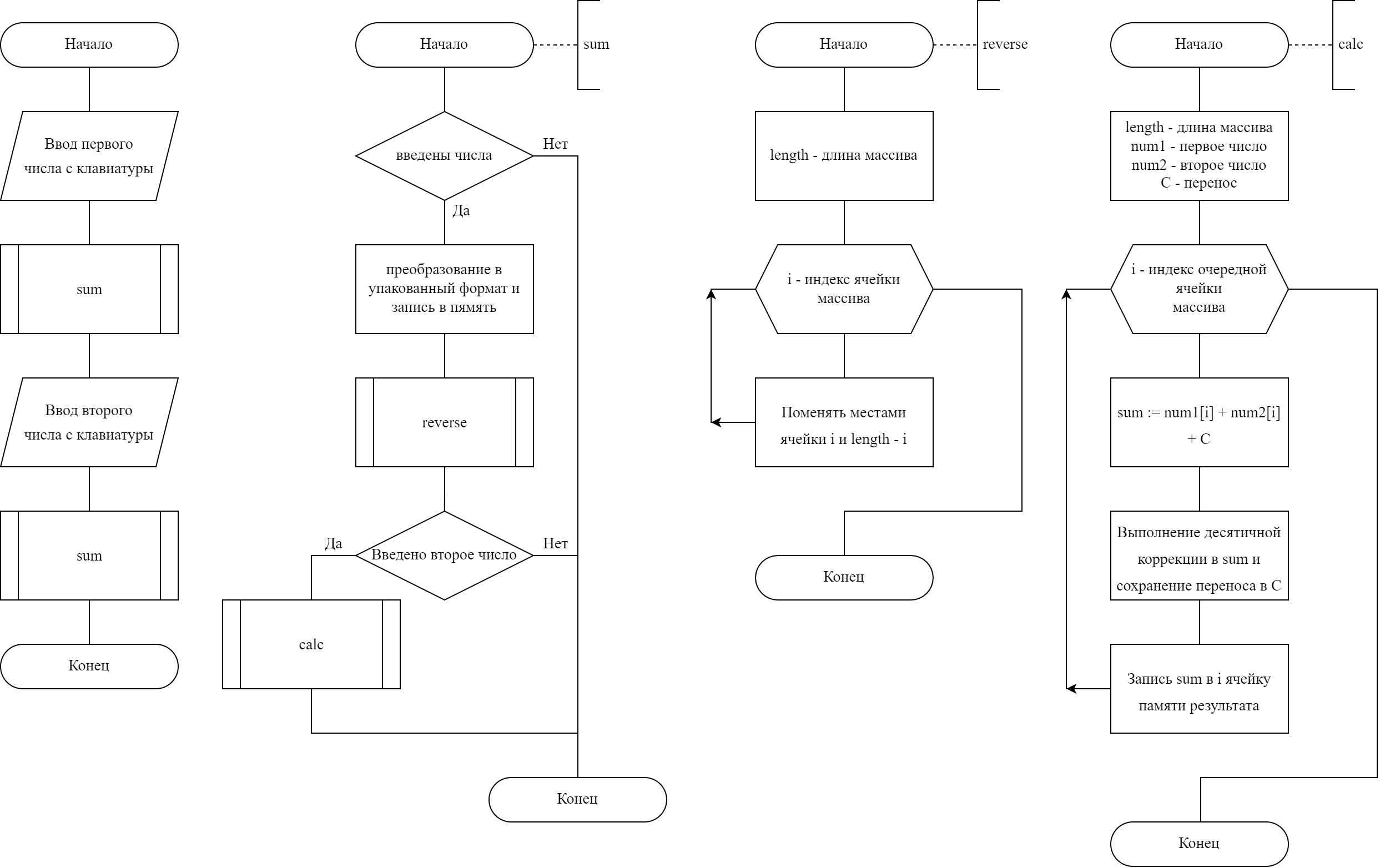


Рисунок 2 – граф-схема алгоритма основной программы и подпрограммы

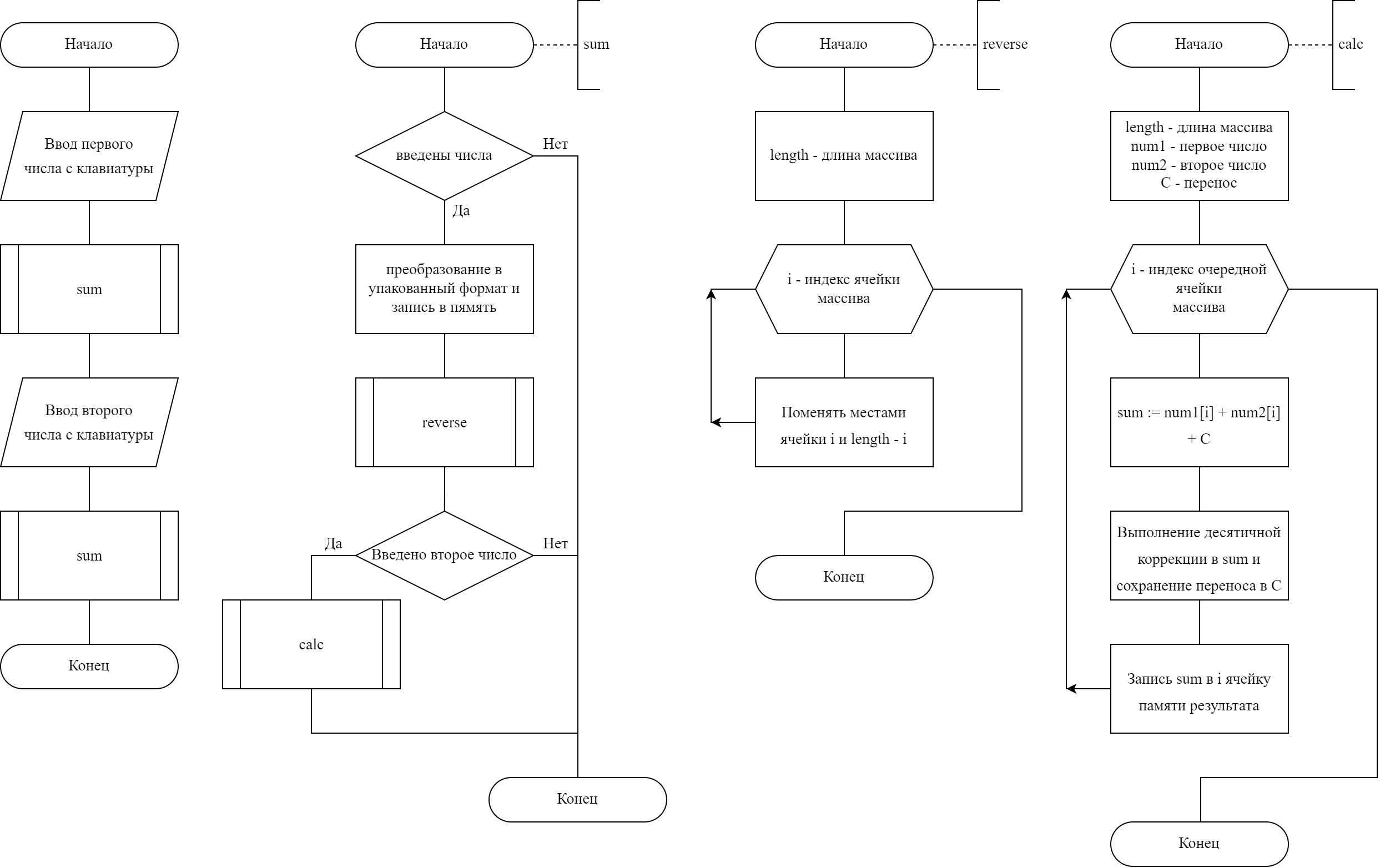


Рисунок 3 – граф-схема подпрограмм

**Текст программы на языке Ассемблера fN8 с комментариями:**

rd #0b111

out 0x01 ; вкл приёма в буфер и формирований запросов на прерывания в строчном режиме

rd #0x20 ; адрес записи

wr r1

wr r0

ei

loop:

nop

jmp loop

correct: ; коррекция при нечётном кол-ве цифр в вводе, r0 начало r1 конец+1 r3 адрес текущей ячейки

wr @r1+

rd r1

sub #1

wr r3

lp:

rd -@r3 ; перенос из i-1 в i

and #0x0F

mul #16

add +@r3

wr @r3-

rd @r3 ; изменение i-1

and #0xF0

div #16

wr @r3

rd r3

cmp r0

jnz lp

jmp input\_end

sum: ; обработчик прерываний клавиатуры

in 0x00

cmp #0x2F

jnc end

cmp #0x39 ; проверка на числа

jc end

sub #0x30

wr r2

sbis 0x02, 1 ; если последний символ в разряде единиц

jmp correct ; откорректировать

mul #16 ; преобр в упакованный формат

wr r2

sbic 0x02, 1

in 0x00

cmp #0x2F

jnc end

cmp #0x39

jc end

sub #0x30

add r2

wr @r1+

input\_end:

sbic 0x02, 1

jmp sum

call reverse ; инвертирование порядка ячеек памяти

end:

iret

reverse: ; r0 начало, r2 кол-во/2, r3 индекс, r4 tmp, r5 левый индекс, r6 правый индекс

rd r1

sub r0

div #2

wr r2

rd #0

wr r3

swp:

rd r0

add r3

wr r5

rd @r5

wr r4

rd r1

sub r3

sub #1

wr r6

rd @r6

wr @r5

rd r4

wr @r6

inc r3

rd r3

cmp r2

jnz swp

rd r1

cmp #0x2F

jnn endr

call calc

jmp exit

endr:

rd #0x30

wr r1

wr r0

ret

calc: ; расчёт результата и вывод в память, r0 счётчик r1 ссылка 1 r2 результат

rd #0

wr r0

wr r2

next:

rd #0x20

add r0

wr r1

rd @r1

wr r2

rd #0x30

add r0

wr r1

rd @r1

sbs 0x10, 0 ; если нет переноса

jmp m1 ; пропустить обработчик переноса

add #1

cb 0x10, 0 ; сбросить перенос

m1:

add r2

daa

jnc m2 ; если нет нового переноса не устанавливаем бит

sb 0x10, 0

m2:

wr r2

rd #0x40

add r0

wr r1

rd r2

wr @r1

inc r0

rd r0

cmp #0x0F

jnn next

ret

exit:

hlt

.org 2

.dw sum

**Распределение памяти:**

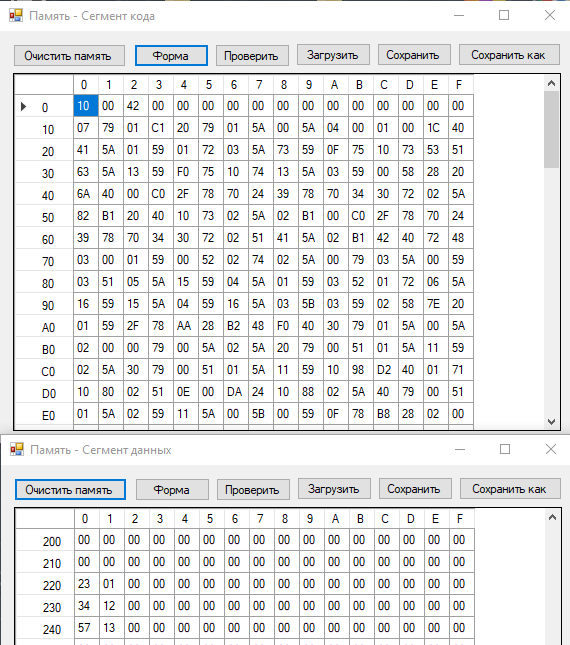
****

Рисунок 4 – Распределение памяти

**Тестирование:**

Тест 1: 123 + 1234 = 1257

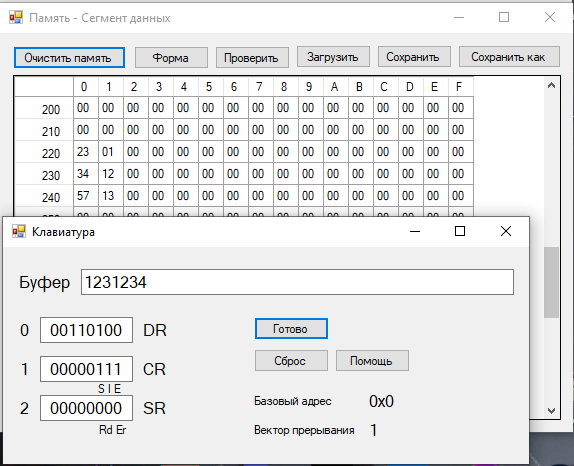
****

Рисунок 5 – Тест 1

Тест 2: 4569 + 891205 = 895774

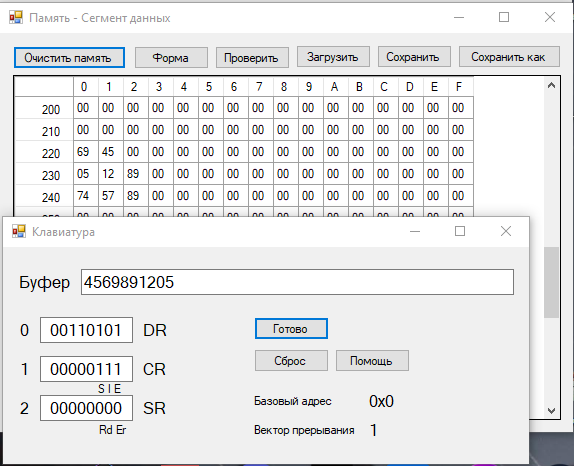
****

Рисунок 6 – Тест 2