Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЕТ

полабораторнойработе№3

по дисциплине: «Алгебраические и логические основы вычислительнойтехники»

натему:«Форматпредставлениячисел cПТвцифровыхпроцессорах»

Выполнил:

студент группы \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_

Принял:

\_\_\_\_\_\_\_\_\_\_\_\_

Пенза,2023

**Ход работы**

**Целые числа**

1. Представил числа **a**=32 и **b**=-77 в формате короткое вещественное(КВ).

**a** = 3210 = 10 00002 = 1,000 0000 0000 0000 0000 0000\*10101

порядок: 111 1111+101=1000 0100

0 100 0010 0000 0000 0000 0000 0000 0000= 4200 000016

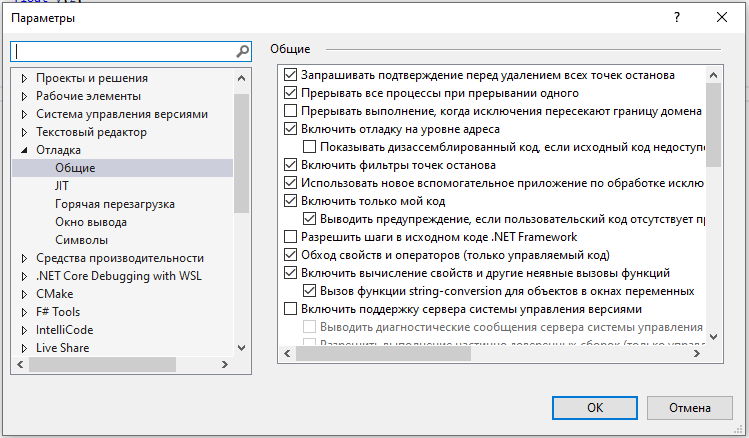
**b** = -7710 = -100 11012 = -1,001 1010 0000 0000 0000 0000 \* 10110

порядок: 111 1111+110=1000 0101

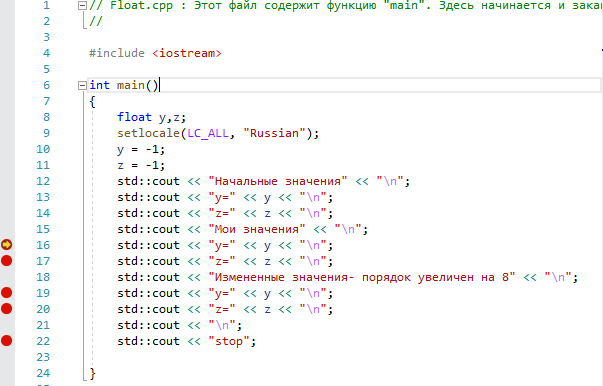
1 100 0010 1001 1010 0000 0000 0000 0000= c29a 000016

1. Для проверки правильности использую отладчик Visual Studio.

Включаю окно «**Памяти**». Чтобы включить окна **Память**, необходимо выбрать параметр **Отладка**>**Параметры** >**Отладка**>**Общие> Включить отладку на уровне адреса.**

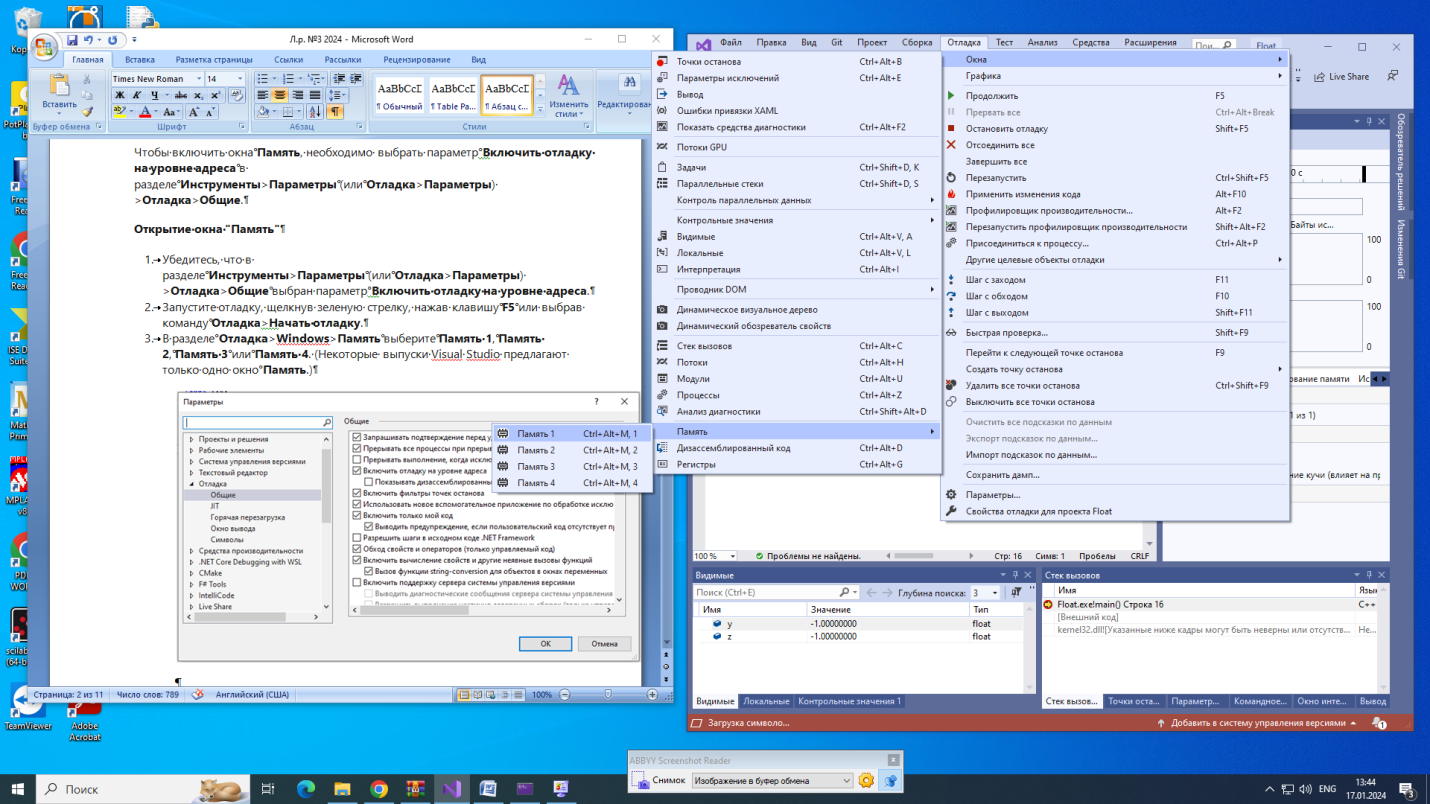


Проект для проверки:

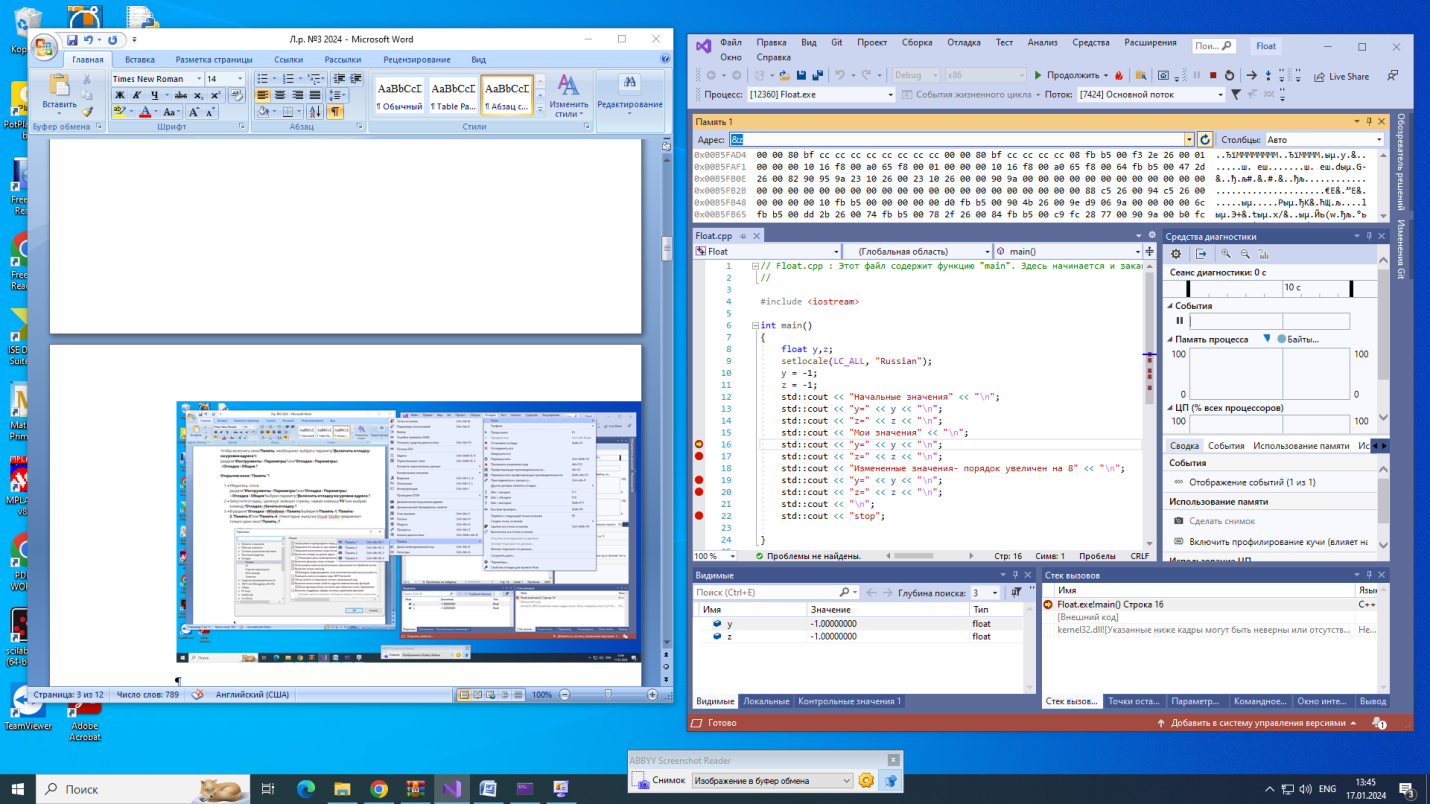


Задаю точки останова в строках кода, где будут проверяться значения переменных. Запускаю отладку, щелкнув зеленую стрелку, нажав клавишу **F5** или **Отладка**>**Начать отладку**.

Показываю окно «**Память»**. Для этого **Отладка**>**Windows**>**Память** выбираю **Память 1**.

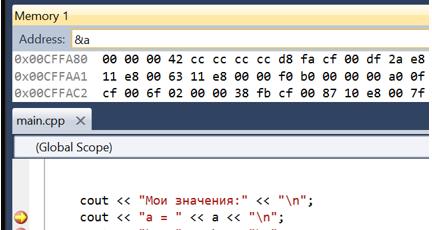


Получил вывод окна памяти:

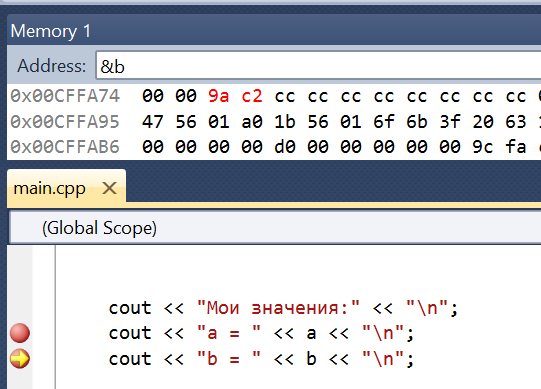


В точке выполненного останова с использованием отладчика, выбрав в памяти адрес переменной **a**, разместил полученные представления чисел в памяти.

Число **a**=4200 000016



Число **b=**c29a 000016



Выполнил программу и получил подтверждение правильности представления чисел **a** и **b** в формате КВ с ПТ.



1. Увеличил порядок каждого операнда на 10002.

**a**: 1000 0100 + 1000 = 1000 1100

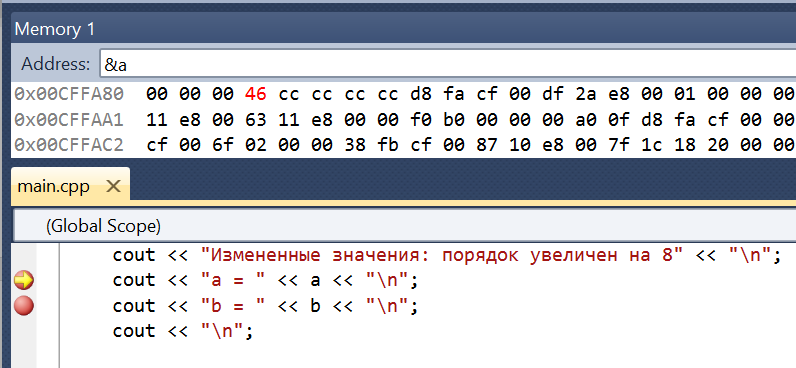
0 100 0110 0000 0000 0000 0000 0000 0000= 4600 000016

**b**: 1000 0101 + 1000 = 1000 1101

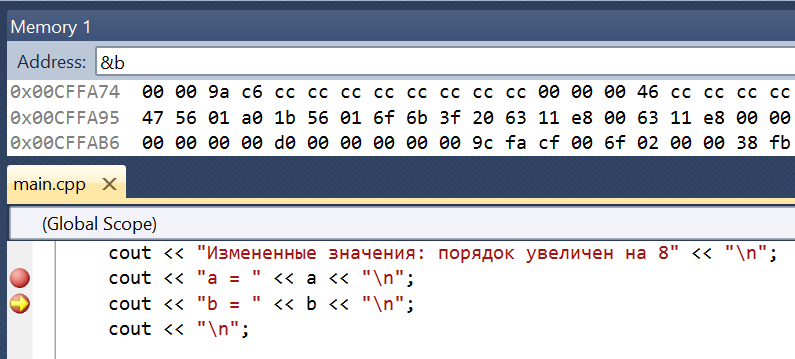
1 100 0110 1001 1010 0000 0000 0000 0000= C69A000016

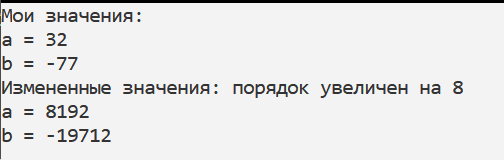
1. Разместил полученные представления чисел в памяти.

**a**:



**b**:





1. Проверяем результат переводом из КВ в десятичную систему счисления.

Для **a**:

0 100 0110 0000 0000 0000 0000 0000 0000

мантисса + «скрытый бит» = 12

порядок = 100011002 – 111 11112= 11012 = 1310

**a** = 12 \* 1013 = 10 0000 0000 00002 = 200016 = 819210

Для **b**:

1 100 0110 1001 1010 0000 0000 0000 0000

мантисса + «скрытый бит» = -1,00110102

порядок = 100011012 – 111 11112 = 11102 =E16 = 1410

**b** = -1,00110102 \* 1014 = 1001101000000002 = -4D0016 = -19712 10

**Дробные числа**

1. Представил числа **c**=0,32 и **d**=-0,77 в формате КВ.

**c** = 0,3210 = 0.0101 0001 1110 1011 1000 01012 =

= 1,010 0011 1101 0111 0000 1010 \* 10-10

Порядок: 111 1111 – 10 = 1111101

0 011 1110 1010 0011 1101 0111 0000 1010 = 3EA3 D70A16

**d** = -0,7710 = -0.1100 0101 0001 1110 1011 10002 =

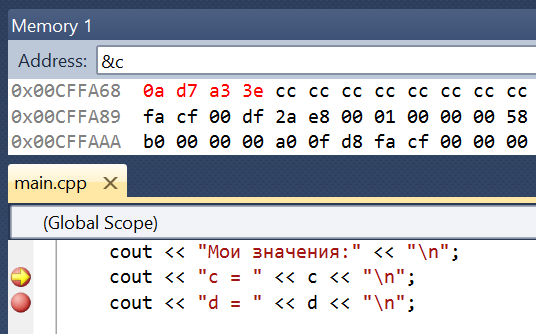
= -1,1000 1010 0011 1101 0111 0000 \* 10-1

Порядок: 111 1111 – 1 = 111 1110

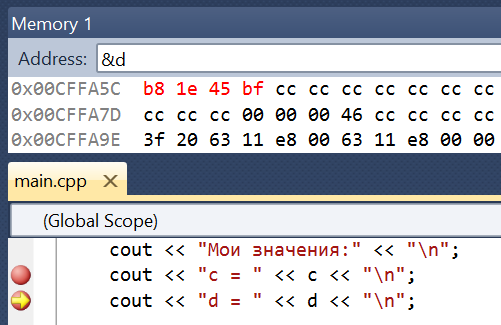
1 011 1111 010001010001111010111000 = BF45 1EB8 16

1. Разместил полученные представления чисел в памяти.

**c**:



**d**:





1. Увеличил порядок каждого операнда на 10002.

**c**: 111 1101 + 1000 = 10000101

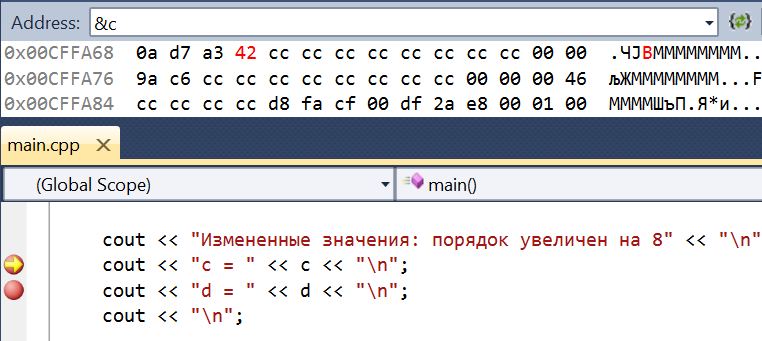
0 1000010 1010 0011 1101 0111 0000 1010 = 42A3D70A16

**d**: 111 1110 + 1000 = 10000110

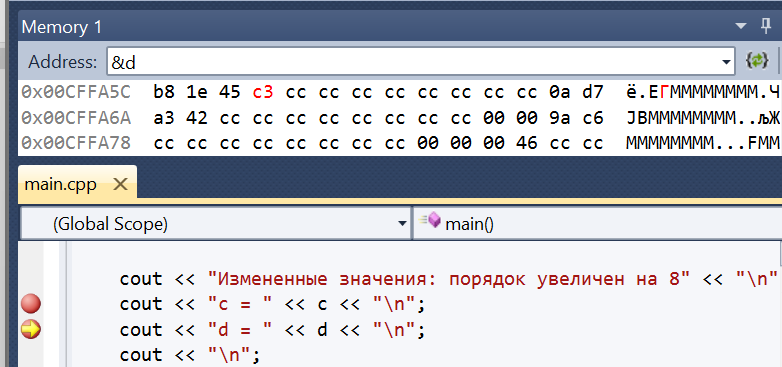
1 1000011 010001010001111010111000 = C3451EB816

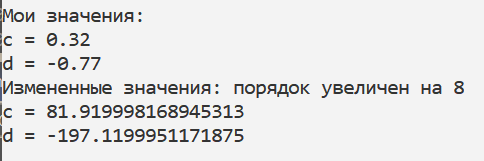
1. Разместил полученные представления чисел в памяти.

**c**:



**d**:





1. Проверяем результат переводом из КВ в десятичную систему счисления.

**c**:

0 1000010 1010 0011 1101 0111 0000 1010

мантисса + «скрытый бит»= 1,010 0011 1101 0111 0000 10102

порядок = 1000 01012 – 111 11112 = 1102 = 610

1,010 0011 1101 0111 0000 10102 \* 106 =1010 001,1 1101 0111 0000 10102 = 51.EB85 16= 81.919998168945312510

Получилось расхождение в младших разрядах из-за того, что результат выводится в формате ДВ, а не КВ (первые восемь десятичных цифр в этом случае должны совпасть, что и видно).

**d**:

1 100 0011 0100 0101 0001 1110 1011 1000

мантисса = -1,100 0101 0001 1110 1011 10002

порядок = 1000 01102 – 111 11112 = 1112 = 710

-1,100 0101 0001 1110 1011 10002 \* 107 = -1100 0101, 0001 1110 1011 10002 = -C5.1EB8 16= -197.1199951171875 10

1. Представил числа **f**=32,77 и **g**=-77,32 в формате КВ.

**f** = 32,7710 = 100000.11000101000111102 = 1,00000110001010001111011\* 10101

Порядок: 111 1111+101=1000 0100

0100 0010 000000110001010001111011= 4203 147B16

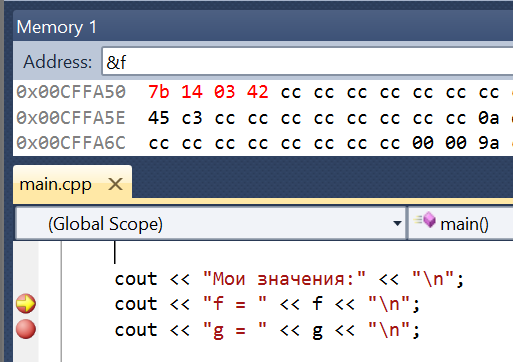
**g** = -77,3210 = -100 1101.0101 0001 1110 10112 = -1.001 1010 1010 0011 110 1 0111\*10^110

Порядок: 111 1111+101=1000 0101

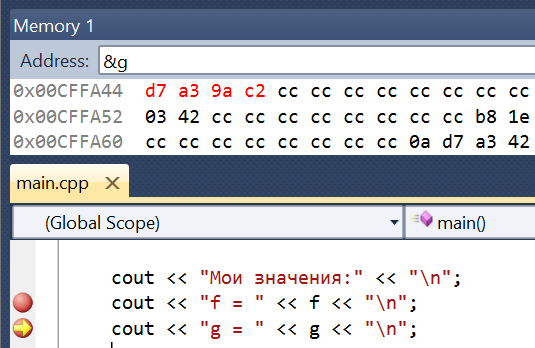
11000010100110101010001111010111= C29A A3D716

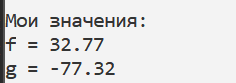
1. Разместил полученные представления чисел в памяти.

**f**:



**g**:





1. Увеличил порядок каждого операнда на 10002.

Порядок: 1000 0100+1000=10001100

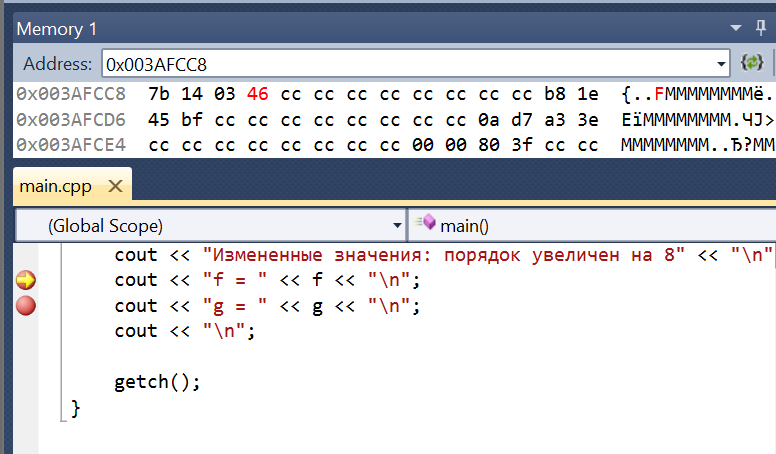
01000110 0000001100010100011110112=4603 147B16

Порядок: 1000 0101+1000=1000 1101

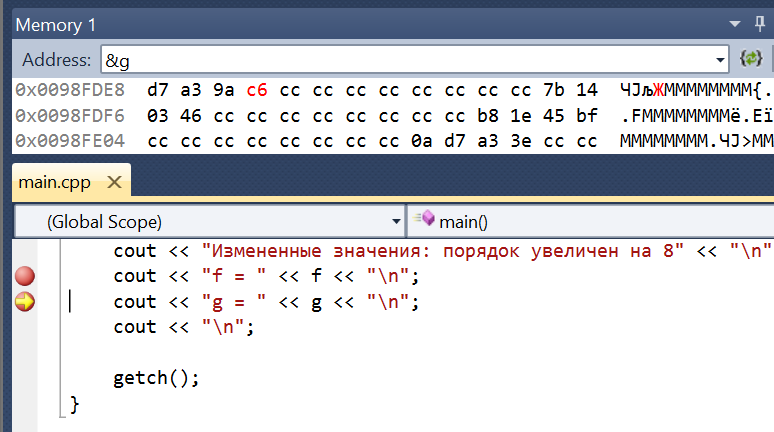
1100 0110 1001 1010 1010 0011 1101 01112=C69AA3D716

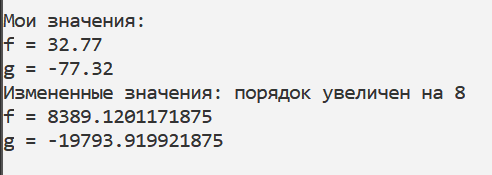
1. Разместил полученные представления чисел в памяти.

**f** :



**g**:





1. Перевел результаты в десятичную систему счисления.

**f**:

01000110 0000001100010100011110112

мантисса + «скрытый бит»: 1,000001100010100011110112

Порядок: 1000 11002 – 111 11112 = 11012 = 1310

1,000001100010100011110112\*1013=10000011000101,00011110112= 20C5.1EC16=8389.120117187510

**g**:

1100 0110 1001 1010 1010 0011 1101 01112

Мантисса: 1,001 1010 1010 0011 1101 0111 2

Порядок: 1000 11012 – 111 11112 = 11002 = 1410

1,001 1010 1010 0011 1101 01112\*1014=1001 1010 1010 001,1 1101 01112 = -4D51.EB8 16 =-19793.919921875 10

**Вывод**:научился представлять целые, дробные и смешанные числа в формате короткое вещественное и переходить от представления числа с ПТ в памяти компьютера к числу в десятичной системе счисления.