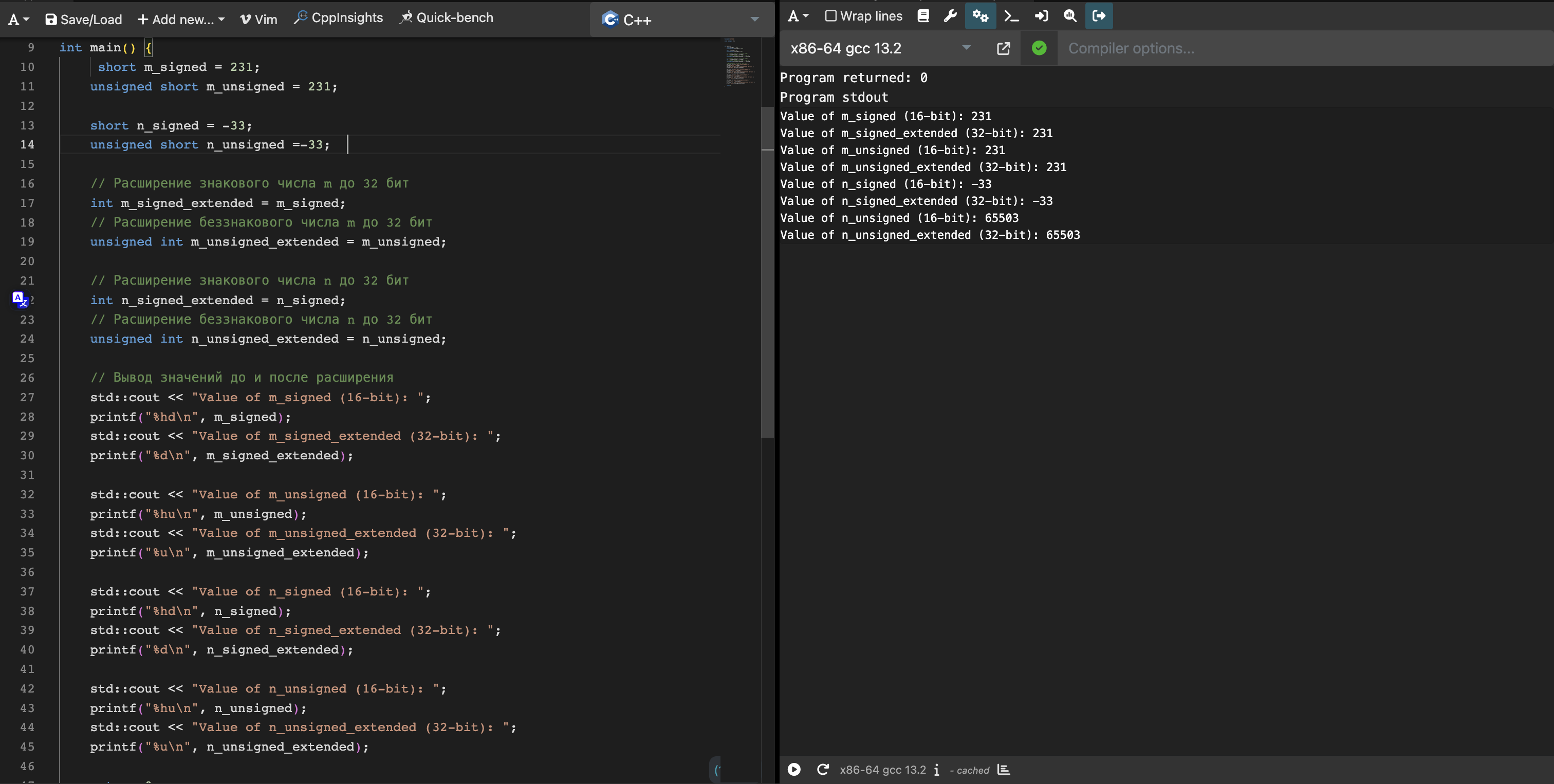
Лаб-3

№1

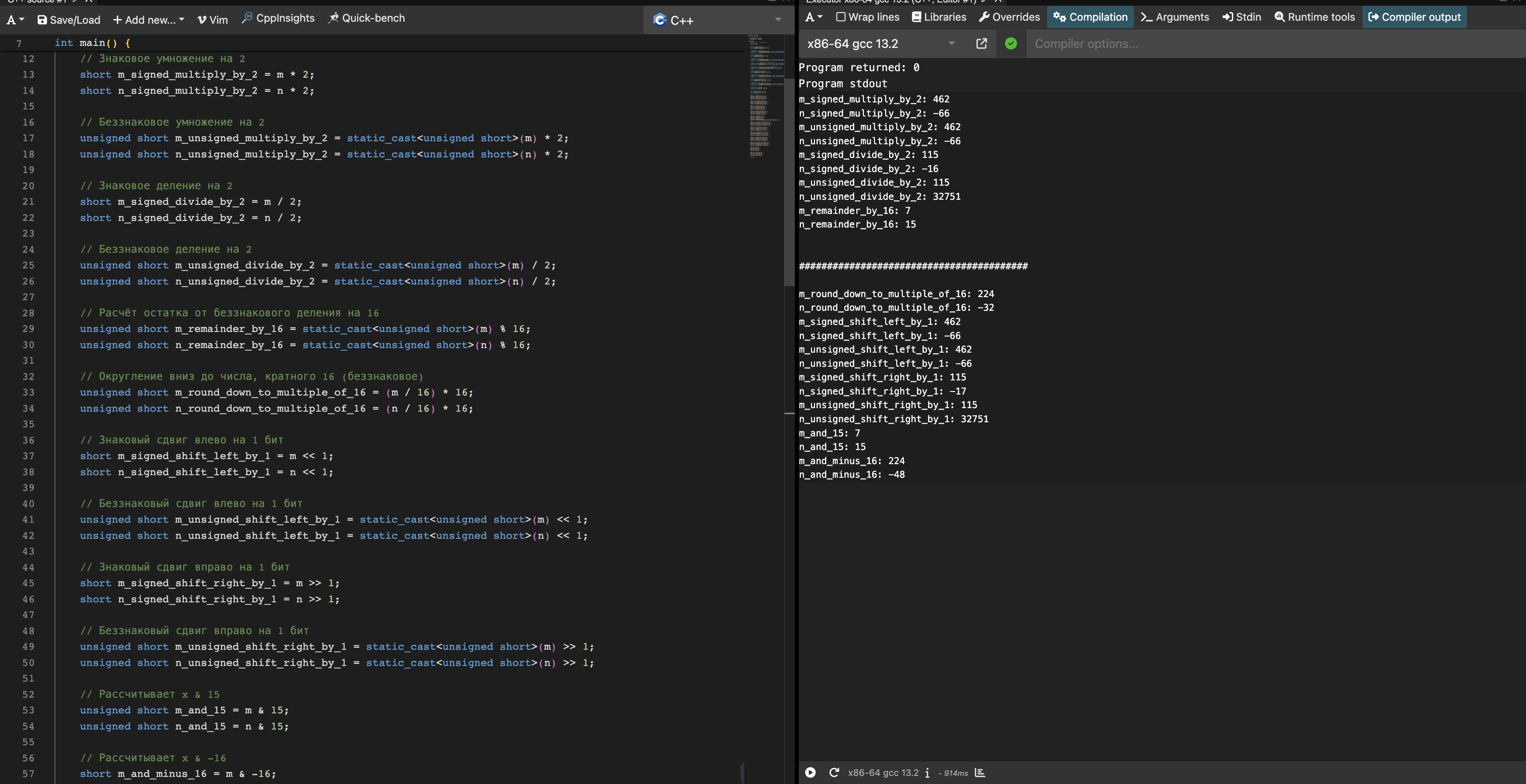
Для использования print16-%hd

print32 - %d

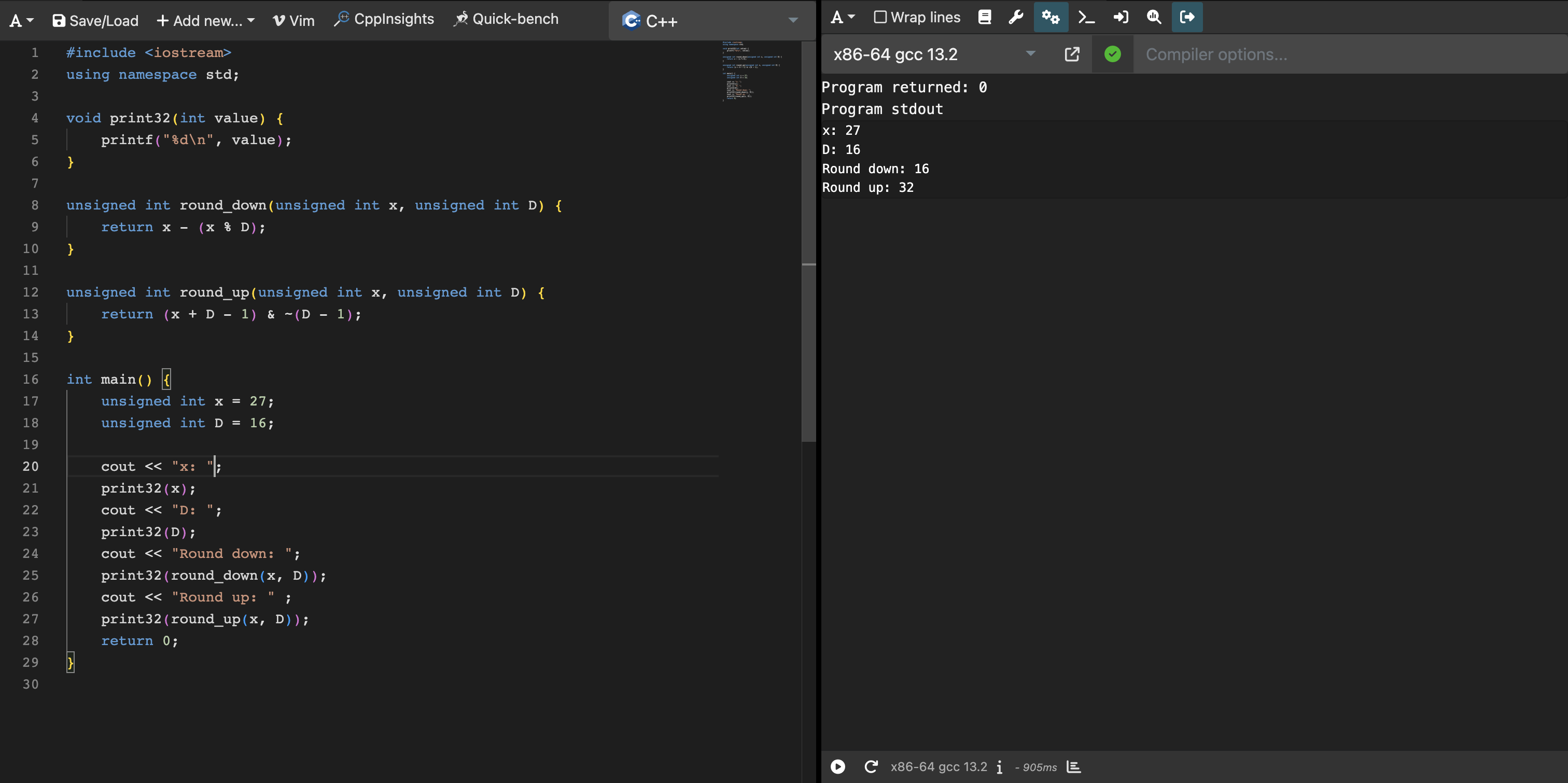


В общем, для значений, которые находятся в пределах диапазона как для знаковых, так и для беззнаковых 16-битных чисел, их расширение до 32 бит не изменяет их значения. Однако, если значение выходит за пределы диапазона 16 бит, то расширение до 32 бит может изменить его значение в случае знакового числа, сохраняя при этом знак, или оставить без изменений в случае беззнакового числа.

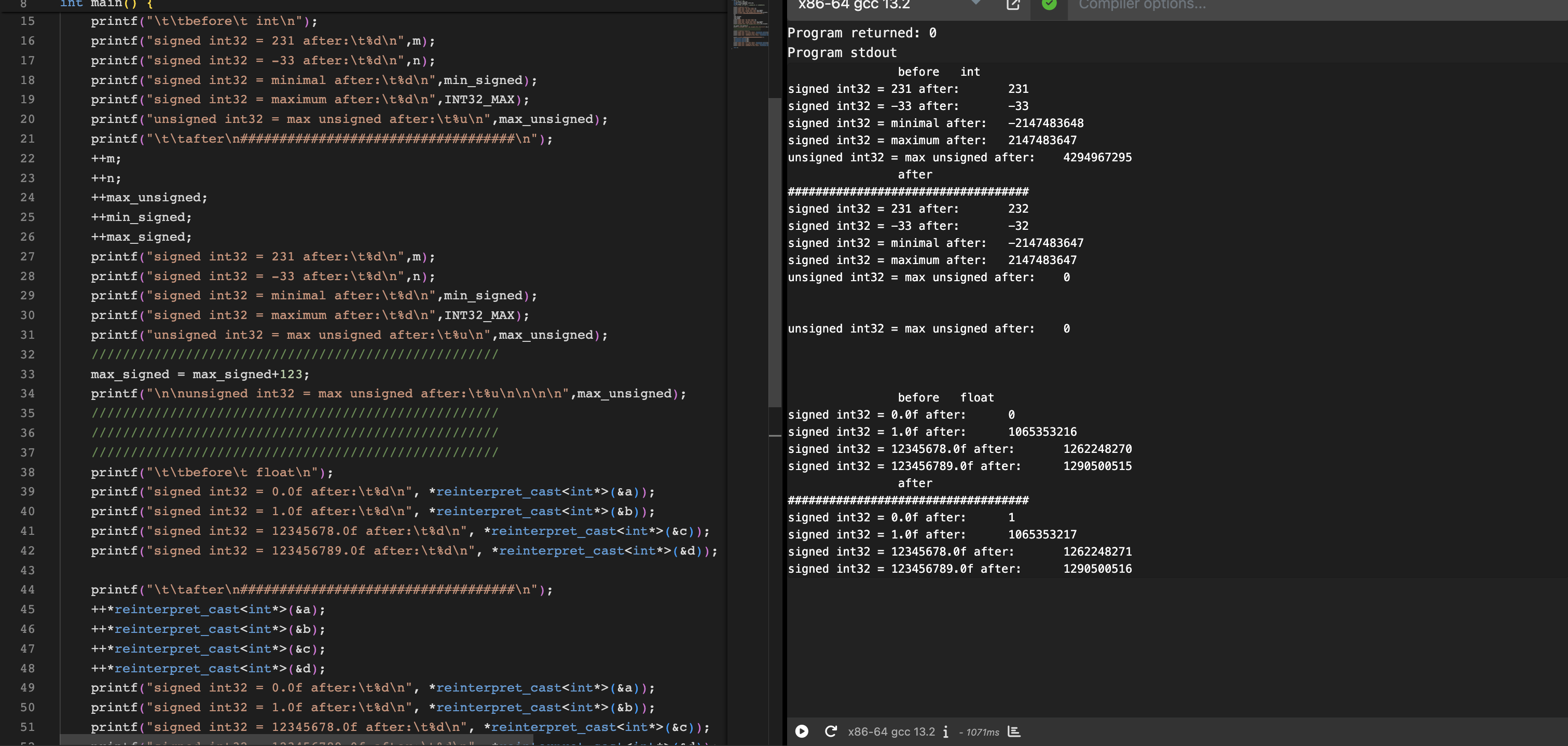
А число -33 превратилось в 65503 так как тип беззнаковый и отминусовалось от 2^16  
№2



№3



№4



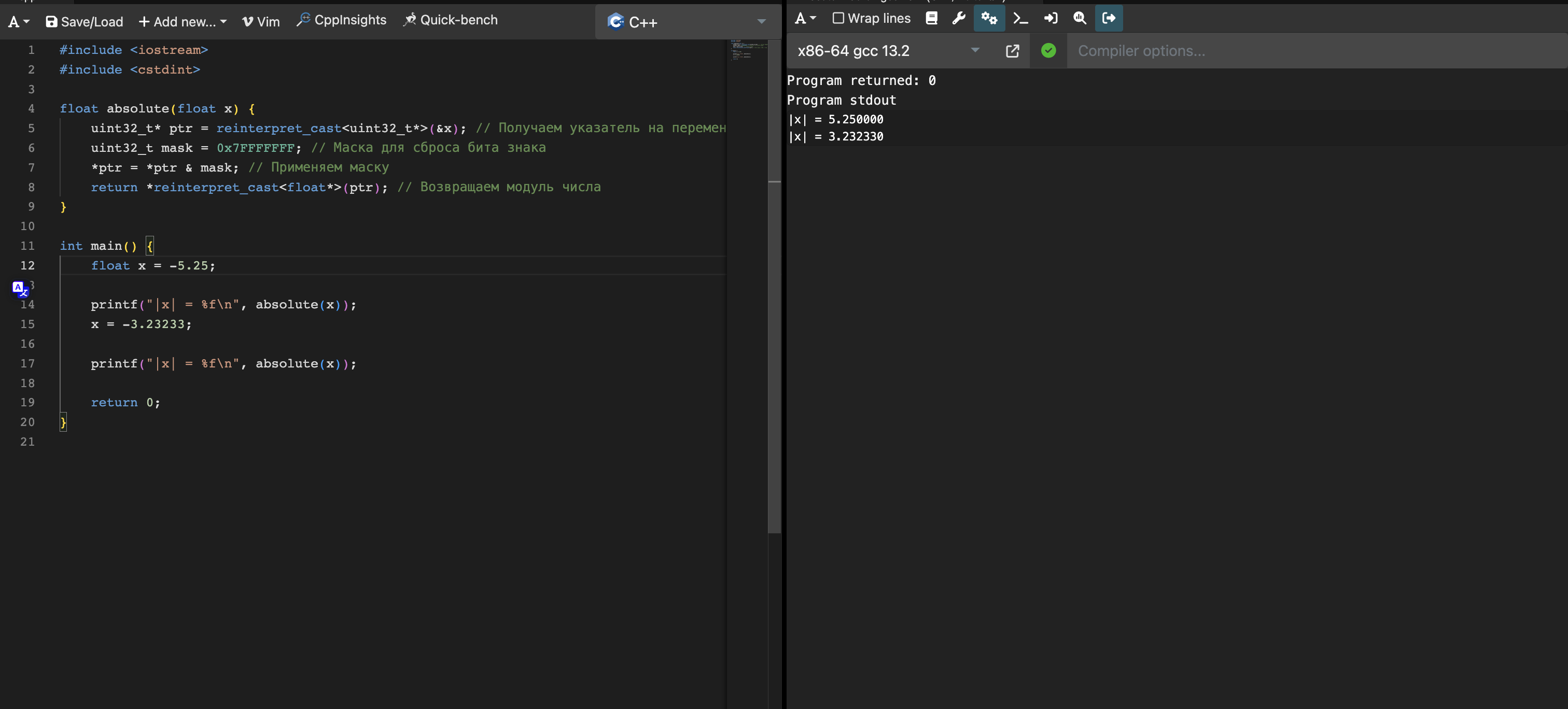
0.0f: В представлении IEEE 754 для числа с плавающей точкой с одинарной точностью (float) ноль будет представлен как нулевой экспонента и нулевая мантисса. Это соответствует целочисленному значению 0.

1.0f: В представлении IEEE 754 для числа с плавающей точкой с одинарной точностью (float) число 1 будет представлено как экспонента равная 0, а мантисса будет представлять единицу среди нулей, что соответствует целочисленному значению 1065353216.

12345678.0f: Значение 12345678.0 в представлении IEEE 754 будет представлено как некоторая экспонента и мантисса, что преобразуется в целое число 1262248270.

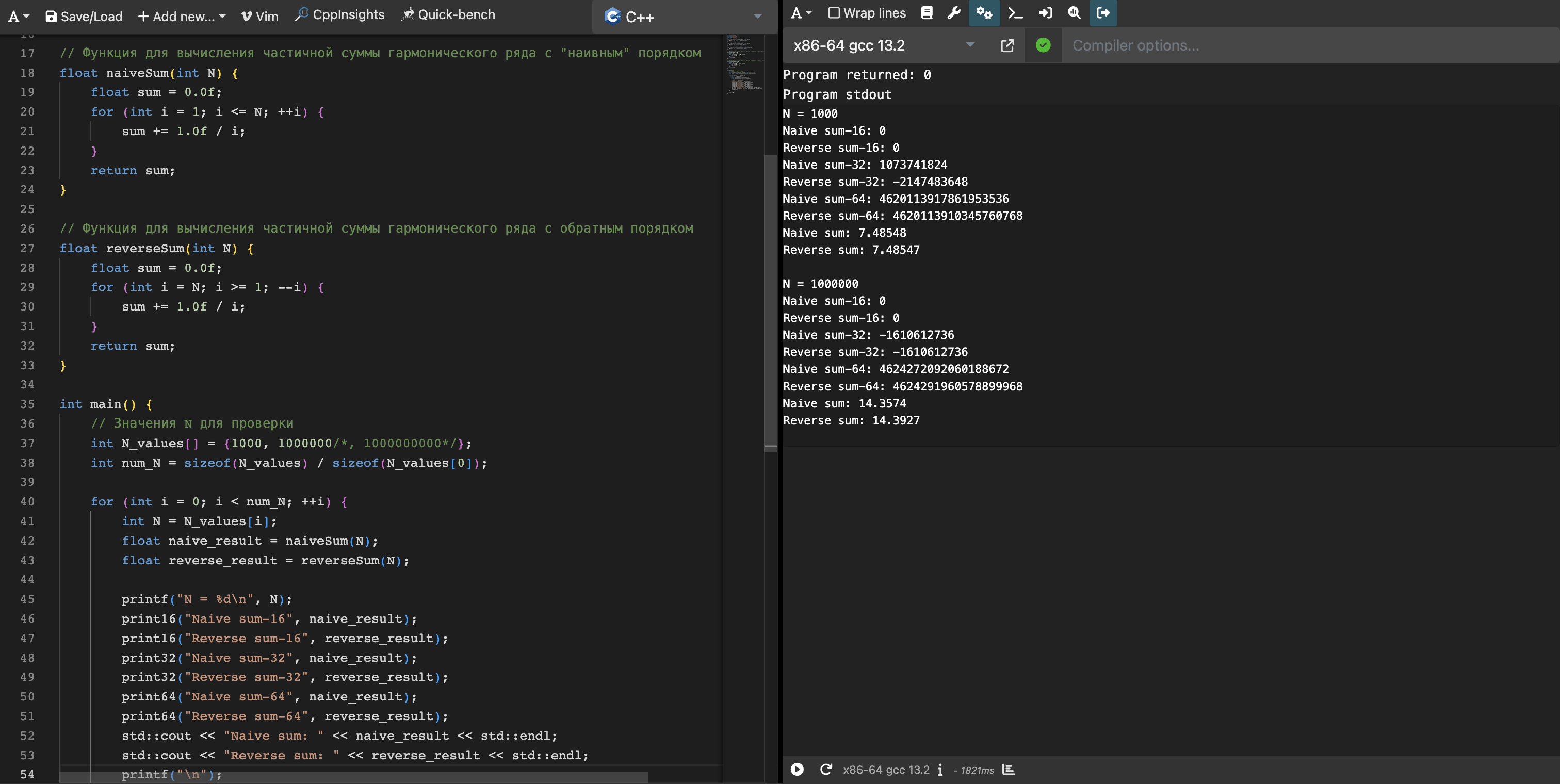
123456789.0f: Это значение в представлении IEEE 754 будет преобразовано в целое число 1290500515.

№5



uint32\_t mask = 0x7FFFFFFF;: Это объявляет переменную mask, которая представляет собой маску для сброса бита знака. Маска имеет вид 0111 1111 1111 1111 1111 1111 1111 1111, что в двоичной форме соответствует значению 0x7FFFFFFF.

№6



Вопросы:  
1. Расширение чисел со знаком и без знака:

Расширение чисел со знаком: Это процесс увеличения разрядности числа со знаком без изменения его значения. Например, если у нас есть 8-битное число со знаком и мы его расширяем до 16 бит, то знак числа сохраняется, а старшие биты заполняются копией старшего бита числа (знаковым битом).

Расширение чисел без знака: Это процесс увеличения разрядности числа без знака без изменения его значения. В отличие от расширения чисел со знаком, все старшие биты заполняются нулями.

Операции расширения могут быть полезны для обеспечения совместимости между различными типами данных, а также для выполнения арифметических операций, таких как сложение или умножение, когда операнды имеют разную разрядность.

2. Логические операции, побитовые операции и сдвиги над строкой битов:

Логические операции (например, логическое И, ИЛИ, НЕ) применяются к каждому биту в двоичном представлении числа и выполняют логическую операцию над этими битами.

Побитовые операции (например, побитовое И, ИЛИ, исключающее ИЛИ) также применяются к каждому биту в двоичном представлении числа, но они выполняют побитовую операцию над этими битами.

Сдвиги (влево и вправо) изменяют позицию битов в двоичном представлении числа. При сдвиге влево все биты смещаются на указанное количество позиций влево, при этом старшие биты, которые выходят за пределы разрядности числа, обнуляются. При сдвиге вправо биты смещаются на указанное количество позиций вправо, а для чисел со знаком старший бит (знаковый бит) расширяется или заполняется в соответствии с его значением для сохранения знака.

3. Представление чисел с плавающей запятой в памяти компьютера:

- В стандарте IEEE 754 предусмотрены различные форматы для представления чисел с плавающей запятой, такие как одинарная точность (float) и двойная точность (double).

- Число с плавающей запятой представляется в виде комбинации мантиссы (значащая часть числа) и экспоненты (степень основания). В формате IEEE 754 мантисса представлена в виде нормализованной дроби, а экспонента представляет собой смещенное значение, которое используется для указания положения запятой.

- В случае одинарной точности (float) число занимает 32 бита (1 бит для знака, 8 бит для экспоненты и 23 бита для мантиссы), а в случае двойной точности (double) - 64 бита (1 бит для знака, 11 бит для экспоненты и 52 бита для мантиссы).

- При работе с числами с плавающей запятой важно учитывать особенности их представления, такие как предельные значения, точность и возможные ошибки округления.