Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

Вариант 3

Выполнила студентка группы КС-30 (Ноль Эльвира Гарриевна)

Ссылка на репозиторий: (https://github.com/Elviranng/Nol-Elvira-KS-30/blob/master/lab4sem2)

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи: 05.06.21

Оглавление

[Описание задачи. 2](#_Toc71231795)

[Описание алгоритма. 2](#_Toc71231796)

[Выполнение задачи. 3](#_Toc71231797)

[Заключение. 14](#_Toc71231798)

# 

# **Описание задачи.**

В рамках лабораторной работы необходимо реализовать 1 из ниже приведенных алгоритмов хеширования:

SHA-2

Провести проверку и построить зависимости скорости расчета хеша в зависимости от размера входных данных для строк длинной (32, 64, 128, 256, 512)

# **Описание алгоритма.**

SHA-2 (Secure Hash Algorithm), в семейство которого входит SHA-256, — это один самых известных и часто используемых алгоритмов хэширования. В тексте подробно покажем каждый шаг работы этого алгоритма на реальном примере. SHA-2 отличается безопасностью (его тяжелее взломать, чем SHA-1) и скоростью.

Три основных цели хэш-функций:

Детерминировано шифровать данные (такой вид шифрования всегда создает одно и то же зашифрованное значение для одного и того же текстового значения);

Принимать ввод любой длины, а выводить результат фиксированной длины;

Изменять данные необратимо. Ввод нельзя получить из вывода.

Алгоритм использует следующие битовые операции:

* ǁ — [конкатенация](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D0%B0%D1%82%D0%B5%D0%BD%D0%B0%D1%86%D0%B8%D1%8F),
* + — [сложение](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5),
* *and* — [побитовое «И»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%8A%D1%8E%D0%BD%D0%BA%D1%86%D0%B8%D1%8F#%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5),
* *xor* — [исключающее «ИЛИ»](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E_2#%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5),
* *shr* (shift right) — [логический сдвиг вправо](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D1%8B%D0%B5_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8#%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%B4%D0%B2%D0%B8%D0%B3),
* *rotr* (rotate right) — [циклический сдвиг вправо](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D1%8B%D0%B5_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8#%D0%A6%D0%B8%D0%BA%D0%BB%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%B4%D0%B2%D0%B8%D0%B3).

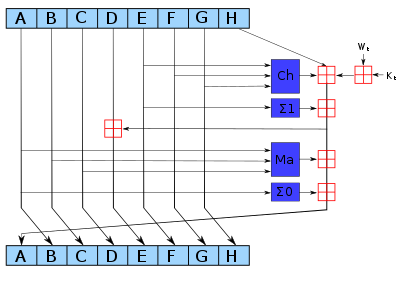
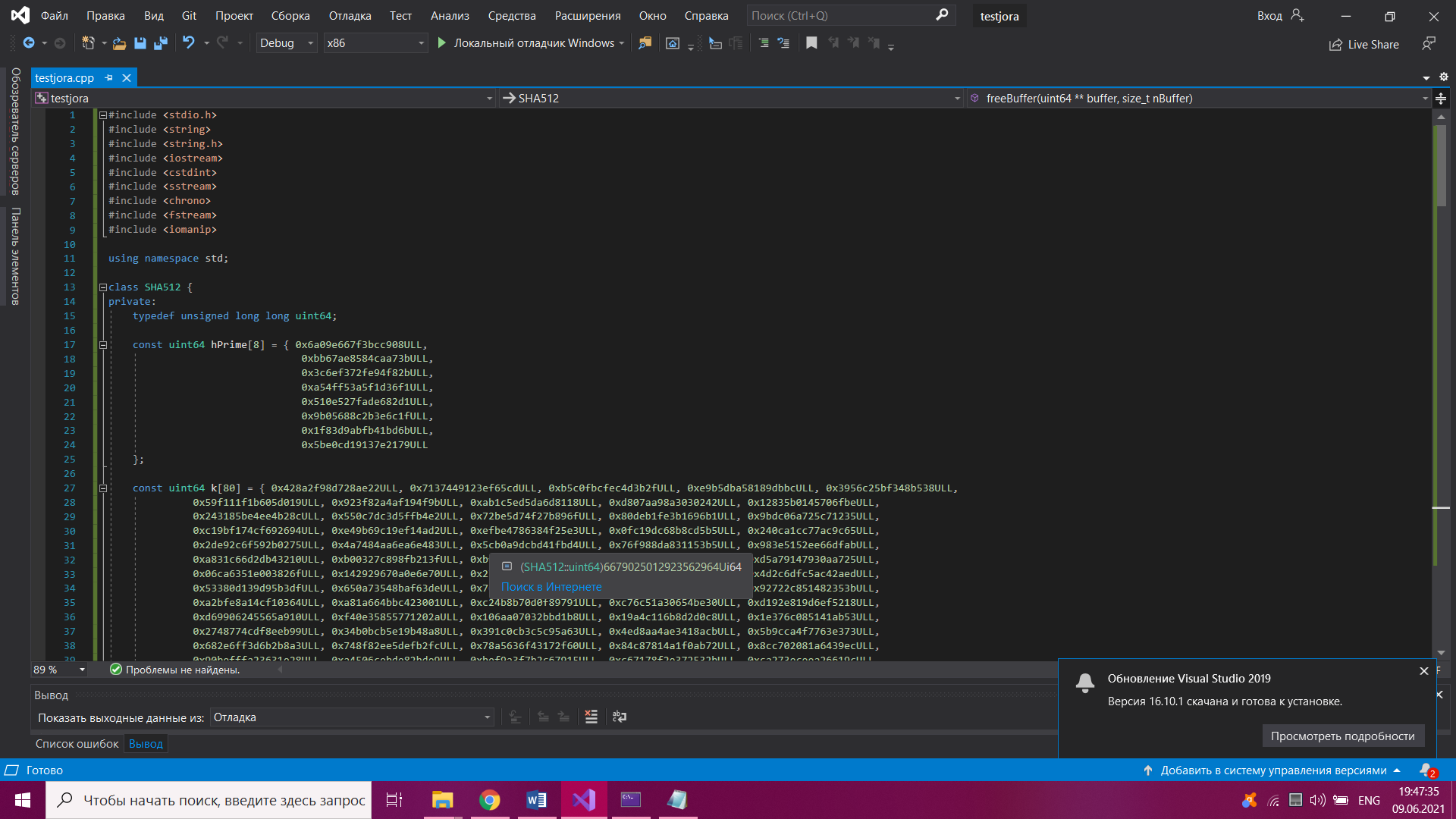


Схема одной итерации алгоритмов SHA-2

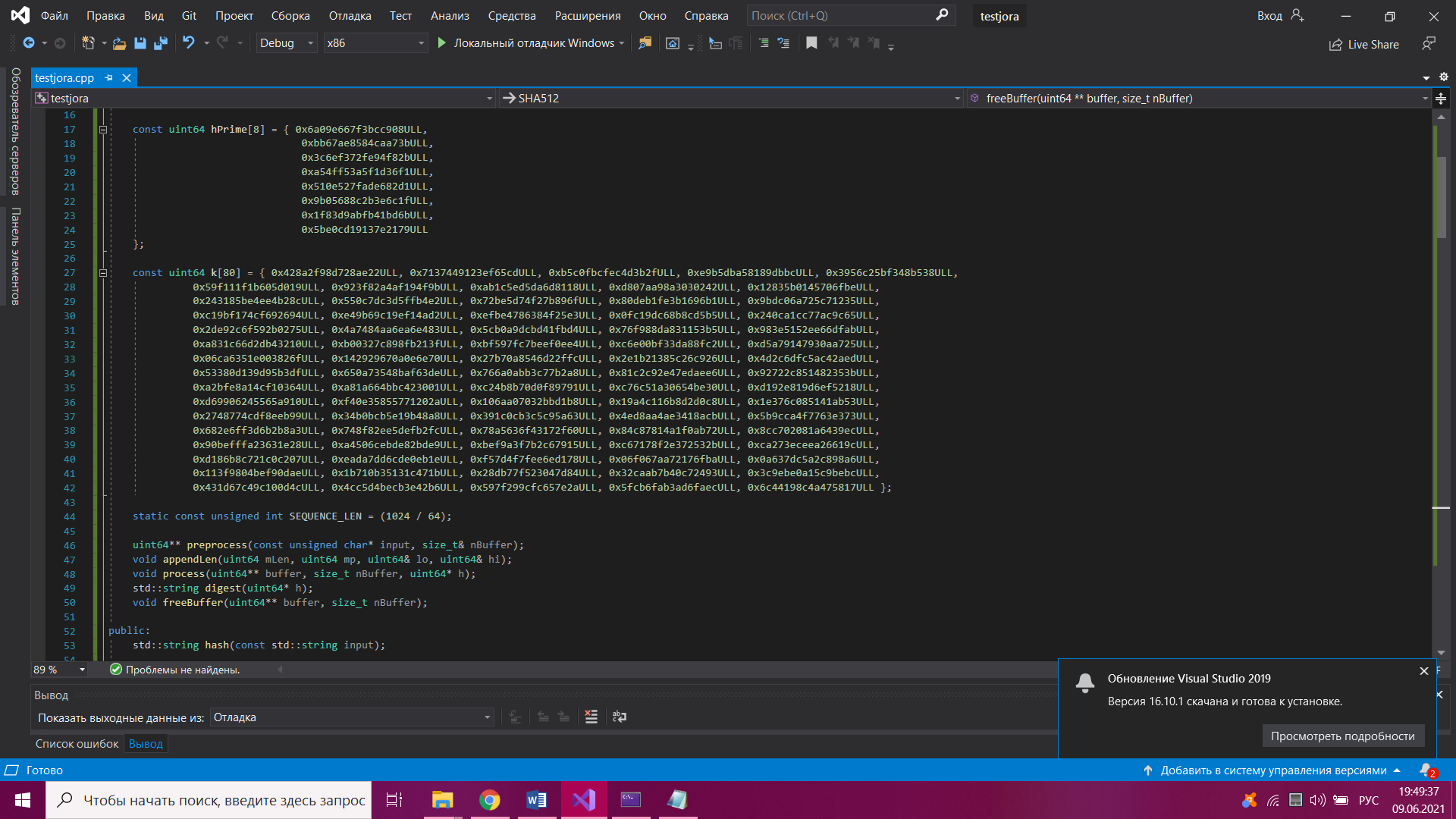
# **Выполнение задачи.**

Преобразуем входные данные в двоичный код, далее ставим единицу и заполняем нулями, пока данные не станут кратны 512, но без последних 64 бит. Для слов или небольших фраз одного блока на 512 бит достаточно.

Далее идет инициализация значений хеша (hPrime):



И инициализация округлённых констант (k):



Далее разбиваем массив с данными на 32битные слова и добавляем ещё 48 таких слов, состоящих из нулей, чтобы суммарно вышло 64 слова.

Изменяем нулевые индексы в конце массива, используя следующий алгоритм:

For i from w[16…63]:

s0 = (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18) xor (w[i-15] righthift 3)

s1 = (w[i-2] rightrotate 17) xor (w[i-2] rightrotate 19) xor (w[i-2] righthift 10)

w [i] = w[i-16] + s0 + w[i-7] + s1

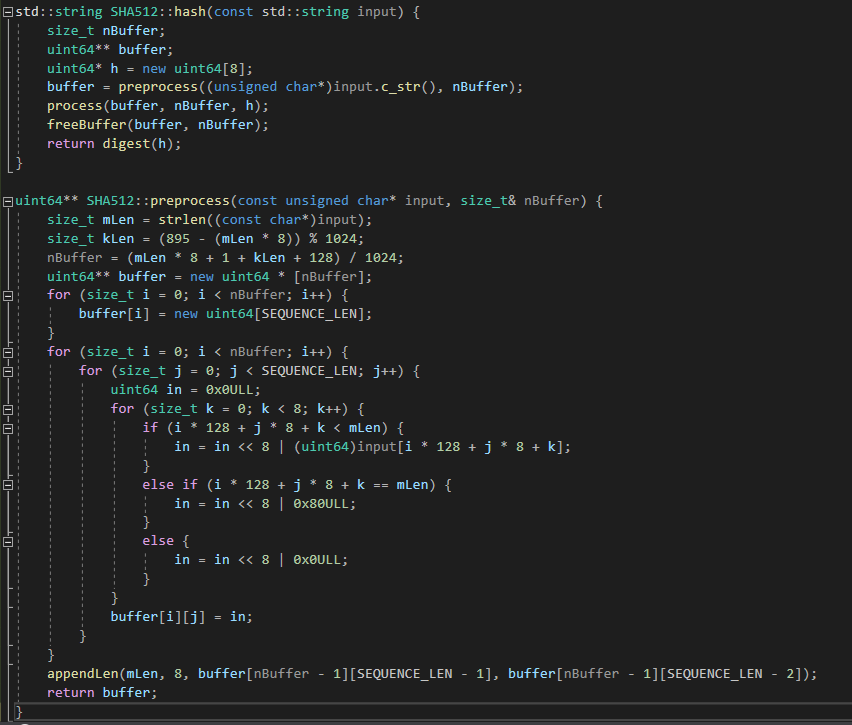
Инициализируем переменные a, b, c, d, e, f, g, h и установим их равными текущим значениям хеша соответственно. h0, h1, h2, h3, h4, h5, h6, h7.

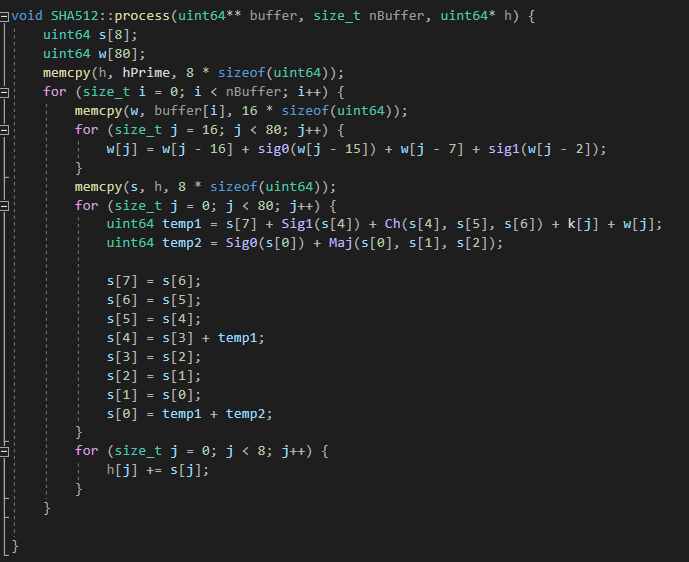
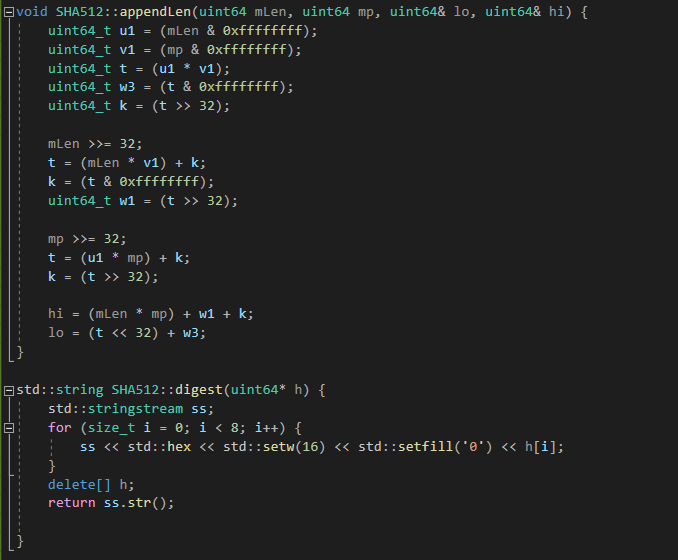
Далее пройдём первую итерацию. Сложение рассчитывается [по модулю](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E) 2^32. После этого все расчеты выполняются ещё 63 раза.

После цикла сжатия, но ещё внутри основного цикла, мы модифицируем значения хеша, добавляя к ним соответствующие переменные a…h. Как обычно, всё сложение происходит по модулю 2^32.

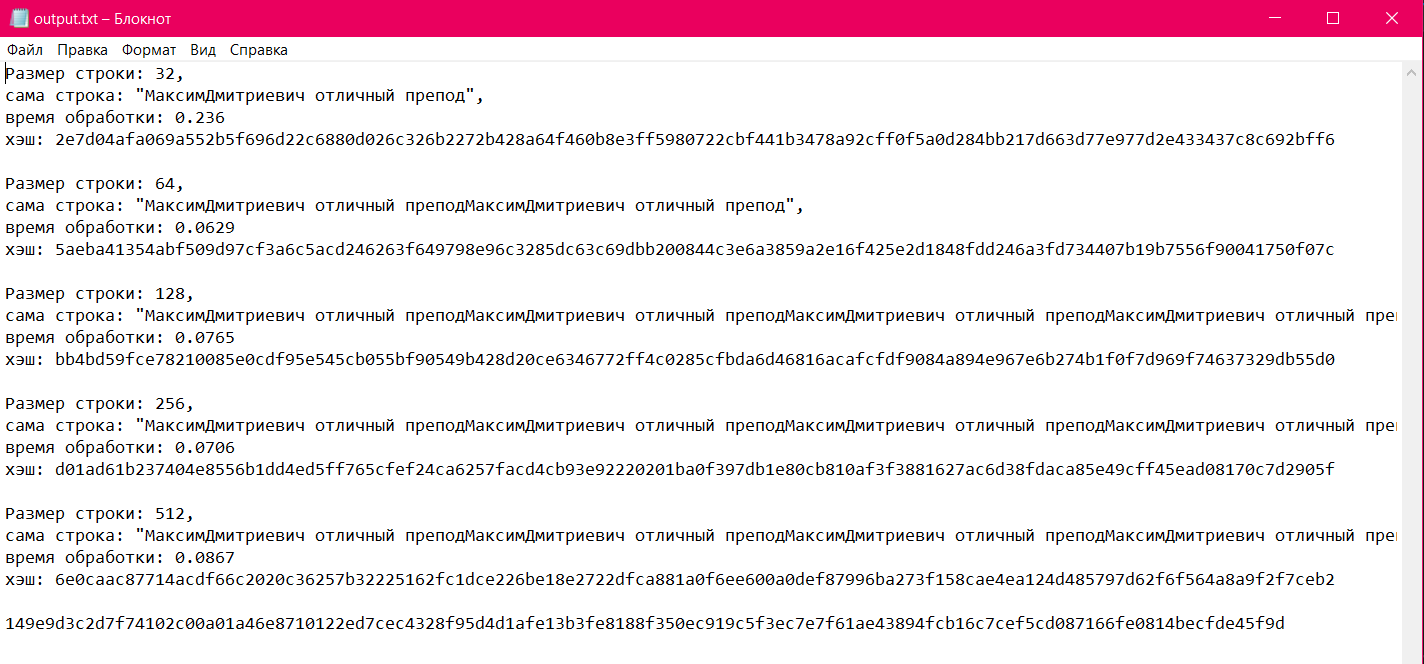
И в последнюю очередь собираем все данные вместе.

Ниже приведу фрагменты программы для хеширования алгоритмом SHA-512:



Результат работы программы:





# **Заключение.**

В ходе лабораторной работы реализовали алгоритм хеширования SHA2 и были выполнены поставленные задчи. Тесты прошли успешно.