Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Криптография»**

Студент: Почечура Артемий Андреевич

Группа: М80-306Б-20

Преподаватель: Борисов Август Валерьевич

Дата: 11.04.2023

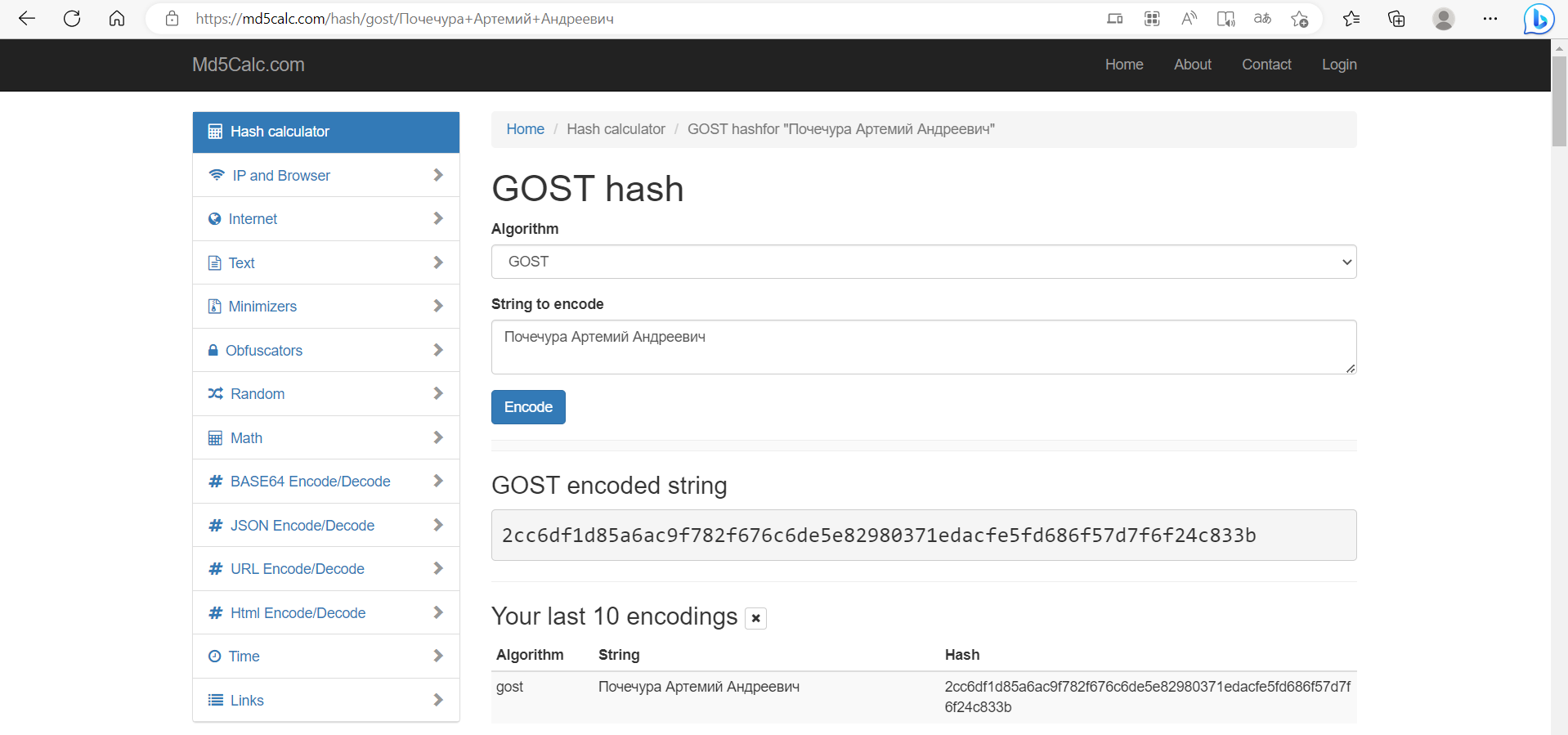
Оценка:

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2023

**1. Генерация варианта**

Получение варианта осуществлялось при помощи сайта 'https://md5calc.com/hash/gost'. На нём была выбрана хэш-функция, соответствующая ГОСТу, и поданы на вход мои ФИО. Из полученной строки, представленной в шестнадцатеричном виде, нужно выбрать два младших разряда. Полученное число и будет являться моим вариантом.



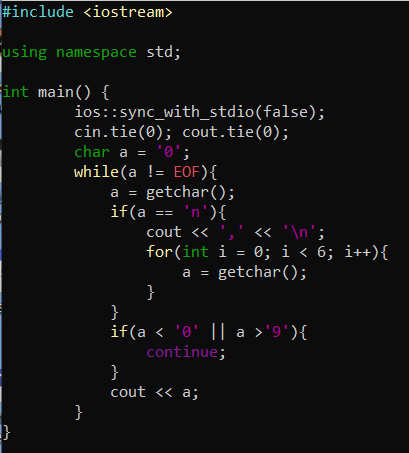
Как видно на скриншоте, мой вариант соответствует следующему числу:

n(3b)=395326274217859209965116210382533368718046735942987071009276138702933739728824277854187187731902371464253636646934323774366698942749289403544659404719427872053573946846381931120576077899695724942600987505840621065174237418164763862267715879364883744349187639964636048875013374611166112340427891206287155929343

**2. Разложение числа на нетривиальные сомножители**

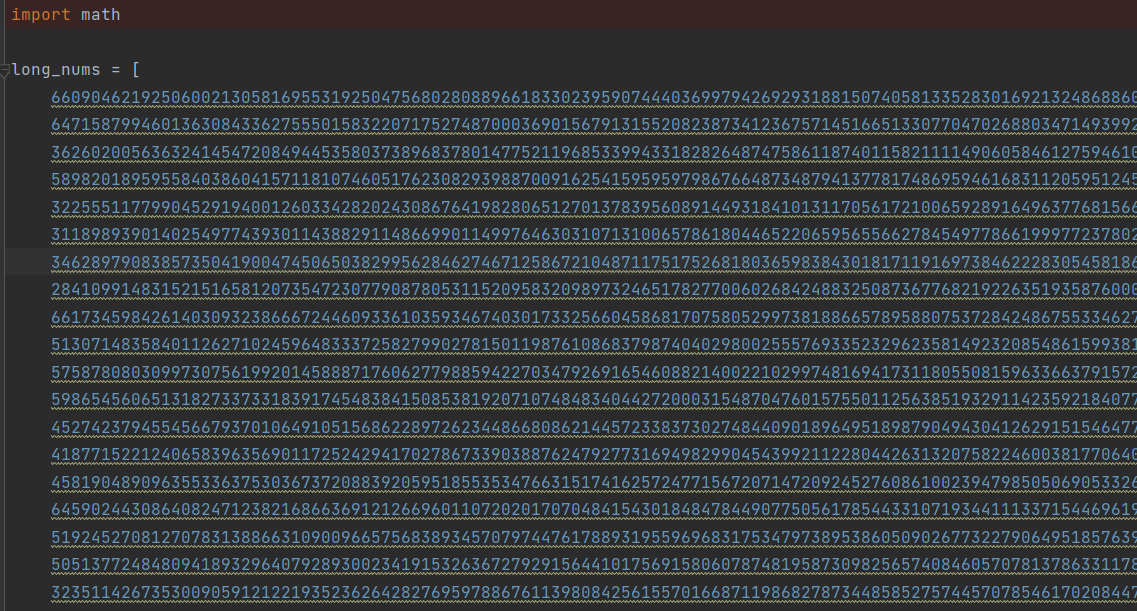
При поиске решения этой части лабораторной мной было выдвинуто предположение, что список всех чисел, данных нам в качестве вариантов лабораторной, был сгенерировать на основе одних и тех же простых чисел. Чтобы это проверить, нужно найти НОД числа, соответствующего моему варианту, со всеми остальными числами из списка.

Для начала этот список потребуется “разпарсить”, чтобы с числами можно было оперативно работать. Делал я это с помощью программы на C++:

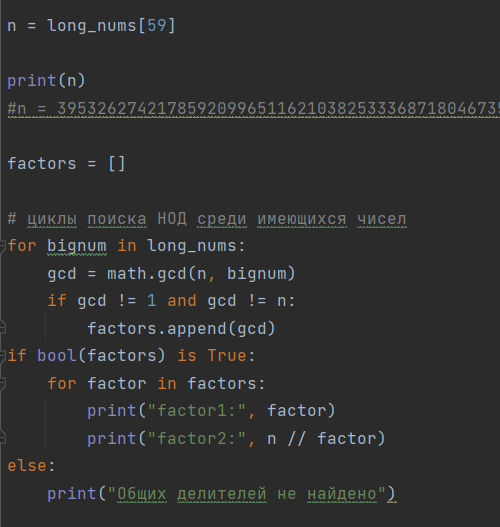




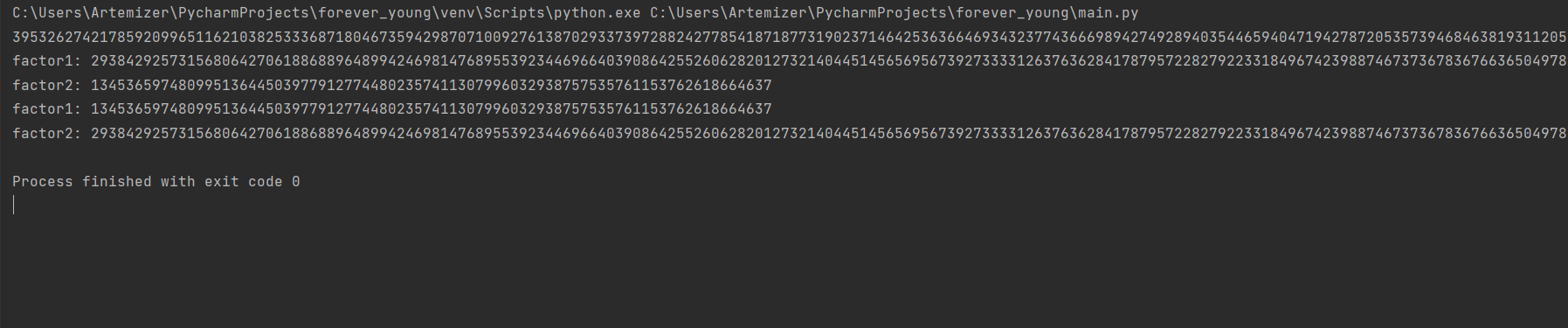
Теперь полученный список можно использовать для реализации программы на Python. В данном языке по умолчанию реализована работа с большими числами, поэтому поиск НОД не предоставит особого труда. Код получился таким:



\*далее идёт список из всех чисел, данных нам в качестве вариантов\*



В ходе работы программа дала следующие результаты:

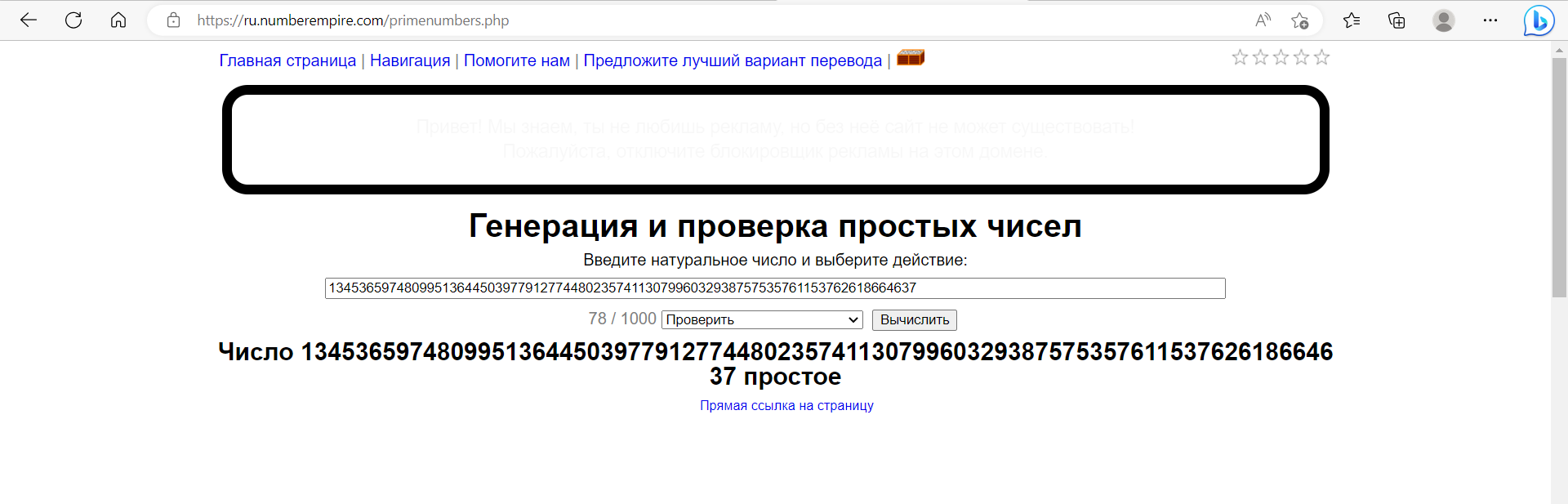


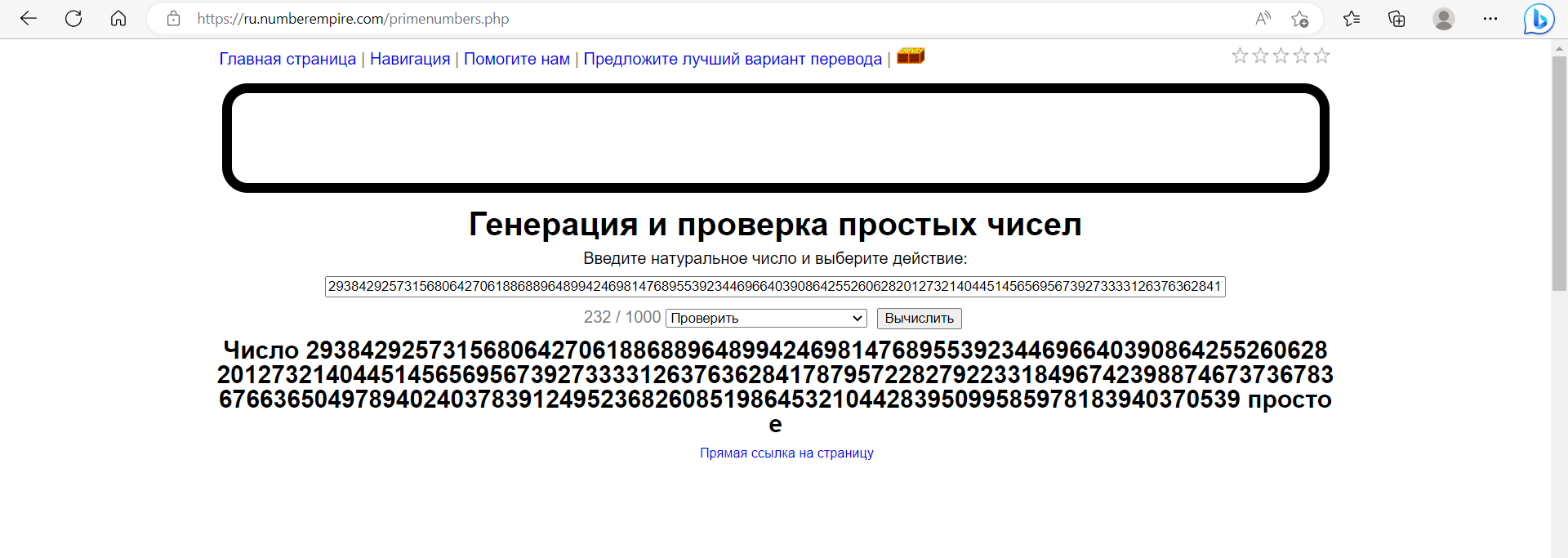
Полученные числа:

134536597480995136445039779127744802357411307996032938757535761153762618664637 и

2938429257315680642706188688964899424698147689553923446966403908642552606282012732140445145656956739273333126376362841787957228279223318496742398874673736783676636504978940240378391249523682608519864532104428395099585978183940370539

Чтобы убедиться, что эти числа простые, воспользуемся сайтом ‘https://ru.numberempire.com/primenumbers.php’. Получаем:





Оба числа являются простыми. Таким образом, разложение числа на нетривиальные сомножители закончено.

**4. Вывод**

В процессе работы над данной лабораторной, мне вновь пришлось факторизировать число на нетривиальные сомножители, однако в этот раз у него было значительно больше разрядов (312 разрядов). Количество нетривиальных сомножителей у данного числа вновь оказалось два, однако количество разрядов у них сильно различается (78 разрядов и 232 разряда). Наличие ровно двух нетривиальных сомножителей подтверждает тот факт, что это число также можно использовать при генерации ключей шифрования. Учитывая то, что ни один известный мне алгоритм не смог разложить данное число (справился только алгоритм с поиском НОД, который задействует и остальные числа из списка вариантов), процесс взлома ключа, сгенерированного с помощью этого числа, перебором становится почти невозможным.