Отчёт по лабораторной работе №3 по курсу

«Криптография»

Выполнил Попов Матвей, группа М8О-308Б-20

*«Если вам действительно нужно разложить число из 309 разрядов на простые множители, то лучше всего обратиться к специалистам в области криптографии или математики, которые могут помочь в решении этой задачи.» © ChatGPT*

# Задание

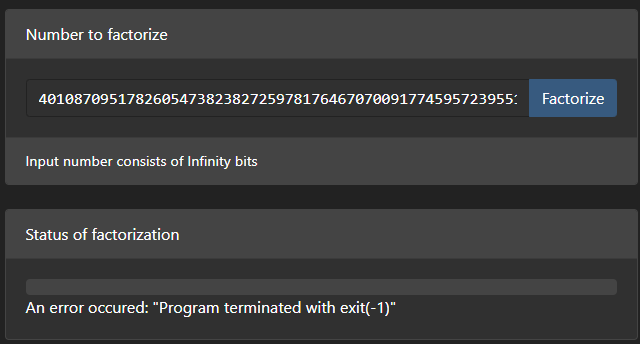
Разложить число на нетривиальные сомножители. Ниже представлены 256 вариантов. Вариант выбрать следующим образом: свое ФИО подать на вход в хеш-функцию, являющуюся стандартом, выход хеш-функции представить в шестнадцатеричном виде и рассматривать младший разряд как номер варианта. В отчете привести подробности процесса вычисления номера варианта.

# Ход работы

Задание выглядит очень знакомо, поэтому сначала я зашёл на уже знакомый мне сайт [convertstring.com](https://www.convertstring.com/Hash/SHA512), вписал в текстовое поле своё ФИО и узнал свой вариант: **B4**. Соответственно, число, которое мне надо разложить:

401087095178260547382382725978176467070091774595723955139318067119016852716047489959471355722273256151644255817878616338859589130668107220552715172981425633887929954782100723684839863673196015575576811236412361999072080407729630058541010398320607440226855035453152182740658040927577508501523918572157928511171

Чтобы разложить такое число, я зашёл на ещё один знакомый мне сайт [cryptool.org](https://www.cryptool.org/en/cto/msieve), ввёл число в текстовое поле, нажал кнопку «Factorize», и столкнулся с неожиданностью:



Удивительно, но это число оказалось слишком большим, поэтому я отправился на поиски специализированного софта для решения моей задачи. Консольная утилита msieve по описанию делала буквально то, что мне было нужно. Я решил установить её на свой компьютер под управлением ОС Ubuntu. Сначала я установил необходимые зависимости, введя в терминале команду:

sudo apt-get install build-essential libgmp-dev libmpfr-dev libmpc-dev

Затем я скачал исходный код программы в папку msieve:

svn co https://svn.code.sf.net/p/msieve/code/trunk msieve

Потом я перешёл в папку msieve:

cd msieve

и запустил сборку исходного кода:

make all

Наконец я запустил программу для разложения моего числа:

./msieve 4010870951782605473823827259781764670700917745957239551393180671190168527160474

89959471355722273256151644255817878616338859589130668107220552715172981425633887929954782100723684839863673196015575576811236412361999072080407729630058541010398320607440226855035453152182740658040927577508501523918572157928511171

То, что я увидел, открыв появившийся файл msieve.log, повергло меня в ужас:

Mon Apr 3 05:47:35 2023

Mon Apr 3 05:47:35 2023

Mon Apr 3 05:47:35 2023 Msieve v. 1.54 (SVN 1046)

Mon Apr 3 05:47:35 2023 random seeds: ffeda411 3b8f73f0

Mon Apr 3 05:47:35 2023 factoring 41548468205797365836521687820371520346496378

807262640592457904803551501105045563694054401077458184109404028058135829623542009593039273975567020311702477385132394167932369753132235424470703827336670645332242350461395779043044704295593184177238952965900154014083979586260763549050618075891853205912909480236739 (308 digits)

Mon Apr 3 05:47:37 2023 no P-1/P+1/ECM available, skipping

Mon Apr 3 05:47:37 2023 commencing quadratic sieve (304-digit input)

Mon Apr 3 05:47:39 2023 using multiplier of 13

Mon Apr 3 05:47:39 2023 using generic 32kb sieve core

Mon Apr 3 05:47:39 2023 sieve interval: 400 blocks of size 32768

Mon Apr 3 05:47:39 2023 processing polynomials in batches of 1

Mon Apr 3 05:47:39 2023 using a sieve bound of 42913231 (1300000 primes)

Mon Apr 3 05:47:39 2023 using large prime bound of 4294967295 (31 bits)

Mon Apr 3 05:47:39 2023 using double large prime bound of 218437776016143360 (51-58 bits)

Mon Apr 3 05:47:39 2023 using trial factoring cutoff of 58 bits

Mon Apr 3 05:47:39 2023 fatal error: poly selection failed

Программа не справилась с разложением такого большого числа. Находясь в отчаянии от отсутствия идей, я решил найти разложение числа из моего варианта путём нахождения наибольшего общего делителя числа из моего варианта с числами из нескольких соседних вариантов. На удивление это сработало, абсолютно случайно попался НОД, не равный 1:

import math

def get\_str\_from\_int(n):

res = list()

while n > 0:

res.append(int(n % 10))

n //= 10

return ''.join(str(x) for x in res[::-1])

nums\_from\_lab\_02 = [219173535765010507529658042240869225453839677822334928192021903994631915023631749028123507427747425017015739257826942730889167625585101786640212951263052154957112501507227133343618307087029774072522203590756620262224654916782730051521366497875819237434957287408367301097012391545495085691103764599446883598793,

224796247945598970030092880075657915636680287244752431438105125628407951295534802043055311849127735411591113617092075139817328294733599266532328228185314019359920998232206828751440127149182785082737725071031422487326726976233129104718458370176861307813547657890967577686542873178905051691563017048461280769477,

434238944874882647573401944705541389075377287180005217169998077075504404193001019512361303478884601345521025576280344280552891923072579436459990741225258910530465715458555257849294574081477663629062758587111051173029132537244759278556191060362295261581898768765525141135112518665527622702928197988374577367347,

301649427662875966271828427347077490078623248239846295898384482496274827712260129561200525144358895951037205707000578034299883714708645885816542354024660940211224148854624761040180933637187628689914432486888577135258145496177998735215575728990705308831347410948947575818170186320260444335342132812204178346601,

401087095178260547382382725978176467070091774595723955139318067119016852716047489959471355722273256151644255817878616338859589130668107220552715172981425633887929954782100723684839863673196015575576811236412361999072080407729630058541010398320607440226855035453152182740658040927577508501523918572157928511171,

510164463347049164286122839139404370659966228722405926421215456891074523886816875597868938896353977658176635547518633847202716920335762930148167533102296524968287348225714852755733046305374456881762382939870515147538350447571042301578964819757861851996557971373279470758887585911225344912753825868505955538159,

531670428231498410644025525196479073399140508813275643027276989584266628562831621129109254665592047379812548187107825064288501019119110608950325349039670107828434421789543151083221488295203070215987925214159617827704081160424780901164820522590847846165340174220677265853736591359252532219147892735677913025011,

450117975294869244123479200495579499834124928507976859010336692997339053524115467089587641405244083241280143758443100810313556252375835280776752692786513944500663521756004522534840509675031224423090921218087174950327994049961199575502759949610756190730834114088477875275845611150846808379689957808171991184901]

v = 4

for x in nums\_from\_lab\_02:

temp = math.gcd(nums\_from\_lab\_02[v], x)

if x != nums\_from\_lab\_02[v] and temp != 1:

print('p =', get\_str\_from\_int(temp))

print('q =', get\_str\_from\_int(nums\_from\_lab\_02[v] / temp))

break

Полученный ответ на задание:

p = 197388488443664937251633715560740748680784664759869599083284568188041142073113

q = 203196801566638282808202826860486028468640006286842426604244600642882824424080026802

4082226220462668024804826484800488488802228268020662084420802448606828484040824286640400802866462022624468488266402864888244848662402068040204624860

# Вывод

Проделав лабораторную работу, я познакомился с программой msieve, а также с увлекательным процессом её установки, проявил находчивость. Конечно мне повезло иметь под рукой число, НОД которого с заданным мне числом являлся ответом, но если бы не везение, то мне бы вероятно никогда не удалось разложить число из варианта B4.