Московский Авиационный Институт

(Национальный исследовательский Университет)

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторная работа №2**

**по курсу «Криптография»**

|  |  |
| --- | --- |
| Студент: | Бердикин Т.А. |
| Группа: | М80-307Б-18 |
| Преподаватель: | Борисов А.В. |
| Оценка: |  |
| Дата: | 09.06.2021 |

Москва

2021

**Содержание**

1. Постановка задачи.
2. Общий метод и алгоритм решения.
3. Демонстрация работы программы.
4. Вывод.

**Постановка задачи**

*Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.*

*Вариант 1:*

n1=35235811807915049318709935514162952710174910616799725550961902052833372235221

n2=1695128485402083763773247025508607781296883851800934596605324477902989989672390098441314233687038522543796524362932674511659084990877094461405769068305253980165481952276151264282270169307424982451349364468884452626363366332792106697498300154504289109043538314722171490851577202002936469515837846884472685701320555954675270470981711883452876152967636160722991943031737727674462234803964546522349706678813412341712703190842025567979822278829254837642753739546649159

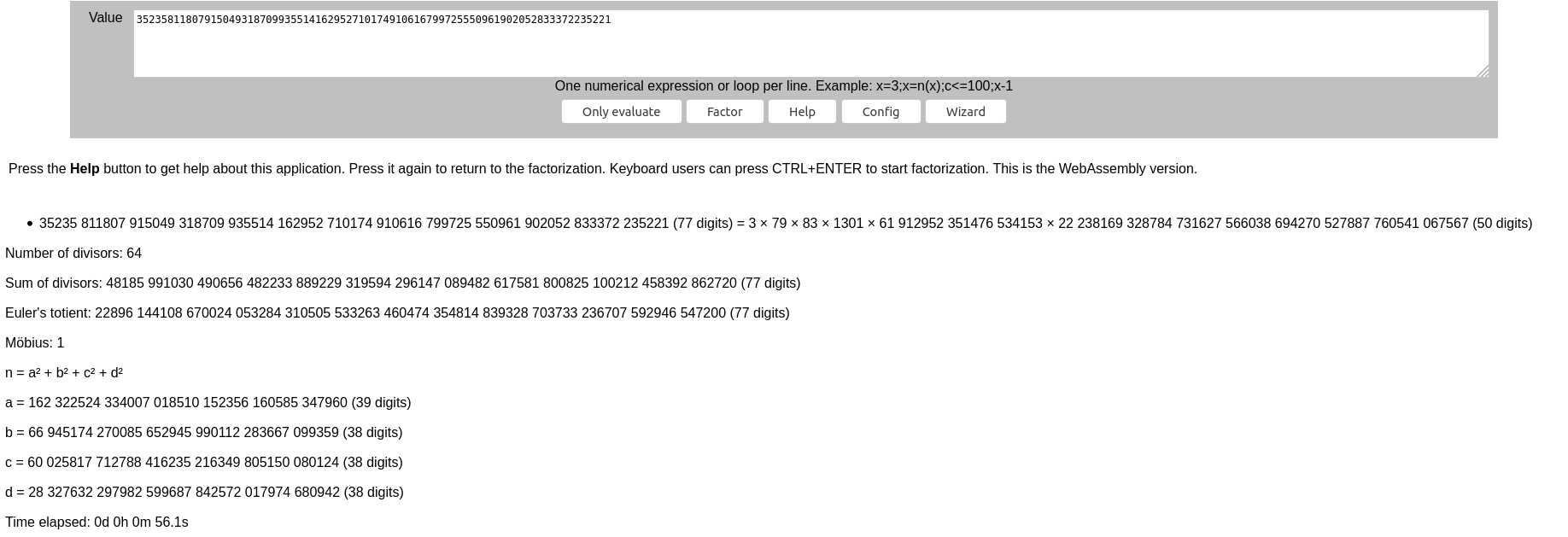
**Методы решения**

Так как ограничений на методы решения не было, я решил разложить первое число с помощью сайта <https://www.alpertron.com.ar/ECM.HTM>.

Второе число не удалось бы разложить таким же путем, во-первых, из-за ограничений на количество знаков, во-вторых, из-за того, что на разложение такого числа понадобилось бы слишком много времени. Конкретно для чисел из вариантов нетривиальный сомножитель можно найти через НОД с числом из другого варианта, а второй – просто делением.

**Полученные результаты**

*Первое число n1:*



*Второе число:*

**tim@tim-GL503VD:~/Desktop/crypto\_tim/1$ python3 lab1.py**

**n2 = 1695128485402083763773247025508607781296883851800934596605324477902989989672390098441314233687038522543796524362932674511659084990877094461405769068305253980165481952276151264282270169307424982451349364468884452626363366332792106697498300154504289109043538314722171490851577202002936469515837846884472685701320555954675270470981711883452876152967636160722991943031737727674462234803964546522349706678813412341712703190842025567979822278829254837642753739546649159**

**from nnumbers is 141824111293255008728651523893019413745439943605780027826534247187938843972987679204447544700784181190531966593770185794553059812672745221625920103329054560319898106605336908519479988189937173062342054739671406574484612829351461016162751493687123552805895463129449138772103341905913170350977470826523504806349**

**factor1 = 141824111293255008728651523893019413745439943605780027826534247187938843972987679204447544700784181190531966593770185794553059812672745221625920103329054560319898106605336908519479988189937173062342054739671406574484612829351461016162751493687123552805895463129449138772103341905913170350977470826523504806349**

**factor2 = 11952329332048507049521674438309669666876305045544046285616098908190253377968711435212264884563859869980038070355418570031673392211221028300427760745838691**

**Код программы**

Lab1.py

#SHORT NUMBERS n1

NUMBERS = [284994967805859272853477327862245466978346919806585432133556769959269315271111,

352358118079150493187099355141629527101749106167997255509619020528333722352217,

119760639583941053725652803731328419697649739176243841021915621242807618608591,

344845228130159226488163571070417679235025139015802019152516926202711846660141,

160769357899975610828199539114109518167531134514190990785144666932076614717841,

274114822339589629024026495441557479713813228028980117869052278950681241194819,

108762353292448487441247663685513658893167646930627178946128889967643172154127,

268887320029090028117214498253204095765884136483366193842361283776500643966781,

123248268911937923199906141216645363665087045422689358104089185316148911496103,

284994967805859272853477327862245466978346919806585432133556769959269315271111,

472379552736871494058143239162622860896965275113543450580272489891667080207763,

361996727456784871855604181056605672088622666207578160811291060873997151708887,

313230894596513941163065516500542159481861849753982064716706926040955753912601,

374456902508739435218273258671224457341348406488533188195528827819627513233269,

61121970174911146319545193754425119520875945215282784640177276523929376501913,

383456614884902466726252731294544234658015390619372835826246625499154384118189,

242587413455689311805941697582103544343444025737930609728129303011307601823551,

181552877565998943910618543225528579935321447209736978912489118450818545230489,

319373613270896663765954115654922624879359841665992852658124487372881123570003,

374456902508739435218273258671224457341348406488533188195528827819627513233269]

#LONG NUMBERS n2

NNUMBERS = [1342124472692680814864696039831657201341930170537490888948185909193404926961223536479474040666460885376378742819197105748650798865879482304771084236256654384273799702336475969647451700146534430052845833553984165082992842025899656567922774484313656763793347645762888379692094136645059343771155043690219662830401572931468290058814304724439824204259980816710796240835274604462076123461698477384443713751284482994607430755155834233283681253132280989394989471961817143,

1695128485402083763773247025508607781296883851800934596605324477902989989672390098441314233687038522543796524362932674511659084990877094461405769068305253980165481952276151264282270169307424982451349364468884452626363366332792106697498300154504289109043538314722171490851577202002936469515837846884472685701320555954675270470981711883452876152967636160722991943031737727674462234803964546522349706678813412341712703190842025567979822278829254837642753739546649159,

1916242087180680156861712994509728052535159091128844805658679025296716559404434664811725619186652725901325774649017594144788360637407178476936316915220758144535681964371311657071750970414707218112222280453951875213591639735019844579642622014874212594838041457800464921182345127496460888250084171815540351211745813542192969624108567504481905290317359415752535077985931507909722167364312980099834023023021212767107040301344392783417575981002593796696074442689507301,

1598756544210860812002683252504666631284038535154979340910964824673923578639226397918134429192737005854188177977059177858243855990803981275665690912975534091041361701843465578101733863479781680791655959578320442108371634048374313524202193198694894536452471646868825144743014452957912743920239954473534374422647748020165306769379396190044599513110393062461302839244356754741065320775011514774723155863731595182892822790709843296375075272651902641460504103291775361,

1250171497372227982026555999675170108947918951378367343470923483104158597216632066586300921566811265776465427395026458151240042366061271512107752586681699923914902061886213022544496783070727061083763996630816279869169194623169255711135422521925444135939014878277515299870536875962948267973899545621728547726545192382593936985574978881305949487523233148677106330650818223443955800622774189936635106363035784698216185461573761714766211607812695281252356674432444279,

1598756544210860812002683252504666631284038535154979340910964824673923578639226397918134429192737005854188177977059177858243855990803981275665690912975534091041361701843465578101733863479781680791655959578320442108371634048374313524202193198694894536452471646868825144743014452957912743920239954473534374422647748020165306769379396190044599513110393062461302839244356754741065320775011514774723155863731595182892822790709843296375075272651902641460504103291775361,

1611765569148804856242867384258680719850010286298191204635154152942043219729044752688614748313611454546572520541736997794001687127300182565577523301374576898637465463079329544247774787283512154983161737116562645744234565727709746364114005583231547967023025414569413122447328040416970845309432217530722433341506166879058135267652737561086239915598233931006566824074208096468336520404693863268533117447729991162579236036416014409092228354404809885779998800076550137,

1417746786978750765038783443201694837693058007147135007928583192144256946704223659049475898042715778235153026085212635256089348105695559658585619676085161346482180413625910718554772936888311138851281270033905970826200499692827568755840858440733991917454028255326174744965696470393644713091831508787116372289467266084564443305079980286049350362289761393863307795187974797187985957533461476088825816395922558727920330066823211210594296302676261707432217348305112187,

1589686907858960532293041950259807409089116075774905924811928369729308516275072914492447303882343619014782861146277474256634429223573812679829885857722514236789773758073602382754296398746760528620467135686904091857677298686613353160501421254539364215543462330529173823254785957892596743971469331053694628704719897511634494080726384449311911326430543608031846181210590808073104043168515626922519393683917981873633828053068169750353137412342101092326814001286079931,

1447056357743040318789862961227509104744799081494678612383291986984923519316446287708049077918224656527429543673229364351887183390807262752423117298211041934655152276599225431751671588895981517419026471542932448198944496908361633132707640798039356570950500607895014150658740782042073630261733525635192524773901831150453706661904186439905176584194604732140346858078193623357352146946016549476780491073212953994660770169348211445199019386069469845306185323206439961,

1262485504020168731000842257581537957328326497522478405002465359648875356810280292244547618070727524417592419776792612058732594852983180148665064058817407866064291179552422627557683886828462061069447032164569235069818669414169882863307032697282802157247652797734392044016320040859257401114524063142894607111829574025600918893253339517061607978068475589931239014683019592991614837523358909806258991077646147246997493894736434495372693444001308001278879395788963879,

1916242087180680156861712994509728052535159091128844805658679025296716559404434664811725619186652725901325774649017594144788360637407178476936316915220758144535681964371311657071750970414707218112222280453951875213591639735019844579642622014874212594838041457800464921182345127496460888250084171815540351211745813542192969624108567504481905290317359415752535077985931507909722167364312980099834023023021212767107040301344392783417575981002593796696074442689507301,

1960344000673448010109966123798259138788312223000110285444138984687043682091918437726564873652655959337927213942829283843615252926281789196372471730892422452230531118265385923148587364956392045025267762404119597838874471039017253236308306374541274375355671500991196394524509192278487473429022067848460150114918996838415401644820324493941862061208584686840594025223786924079444262714095490301772077126395790235999836003971290616988894725373002042174148527448991721,

1688432268535652536976161544225404933352917348466880741646555236080940468369390533777566901374863846088926302716704958253349013465017171687476514345454080829512228091554390695242226222710223271367480753308157792549868681240943730184545304781633011043927327584175094195702062946904306735673354996415907014195550590550472255345961637249641012928019098518193363458415691802243705768503786649882426739768062694678022813527067727278842446759998639312587246098493677573,

1669812028211114876035741593474021802212340044740884701344271270195832085856797314936725609969919892880432470047684464541567265336806788958402625350522072215356887542345091965364412714416147723007824852940439210347535492079930938715301851663504907633271178215986687496281673059795431500200800112337374888657642932011377010779739631990411714885737361714602715397638982646136481630238419481808864438911371804085212946840198558441479176256832689600476668930865222709,

1416908444771934114327236064335695175033855568724514723276090909238902249450761163116179298370097637736609598746978539681190806175023739437824949790203114195447287621192162052863911370030281253311582477023859027984818679108239267600763411891113578181938978341368763677855534685413427437290239276573078365437316891195505584463642669716112936728373088553385902864359218933750627440521470476774178793413097754328106876810009083432628213288672194420754620920548851129,

1510938584302514746068687680359138712084826869531749833816152536107029956694378228665014484809993284680636465045336584670006512692482057168858805251730522412435575537047638759183849437861169582174353100616760861442083338911162982978018654609073487455618344725646474341106448770186119465437436805540314573902315148010605642969399036239279990866481377552631038345038332671300460449150826133047599402952702220438132324240801480483055996850135609380612773088576264939,

1503349990631350512794289684313078245040080234793749288284388102811529318651334186314509247654009172580006457439356152132860210881356046271699329201253058433036162321674305188211885111628223749659497716686816384033178613797568618927173752800692795231622235434934335500496599315357786595208816213489429090618724729416131746965336847080815801599020573311110511374951097277607310799592097578622368423440718164595720091899427036135539095740807639167195995008580910433,

1540622509490817949053524649165981982362710138590145680108367660592300942107455572128874985313317596143790569145958434047726221469334726361045510333507710004199345529052825110132429789384438990706927549015685843329393401520154237372079098668324817011296825302825102856143574580211014626530080895107030337934771878554891597286701169436904639362768184131040154699128645755841829998901472679565730259098946642540287337083284263287432616094697258993945232767013781501,

1626570592384034401231059859408455254810050911431145580773817320385445678597776695068312796145258618012655448521816316108022278762520239267979899184627816793656580906379077745825130933420781919802013703405155696033529555793998359389173755887366857329131343206148632506258546398761725587714083008828347727071434771944964364829767905778186912771898431501076025378480110831840332479020783206206190405100394982218769269156393531604603604142841091039265485070414672259]

import math

v = 1

n = NNUMBERS[v]

print("n2 = ", n)

dividers = []

for num in NUMBERS:

gcd = math.gcd(n,num)

if gcd != 1 and gcd != n:

print("from numbers is ", gcd)

dividers.append(gcd)

for nnum in NNUMBERS:

gcd = math.gcd(n, nnum)

if gcd != 1 and gcd != n:

print("from nnumbers is", gcd)

dividers.append(gcd)

for div in dividers:

print("factor1 = ", div)

print("factor2 = ", n // div)

**Выводы.**

При выполнении данной лабораторной работы мне пришлось изучить кучу статей об алгоритмах факторизации, как быстрых, так и медленных. В любом случае, чем больше число – тем дольше будет ожидание результата. По сути, определённого эффективного алгоритма не существует, каждый подходит для определённых целей и задач, а потому выбрать было достаточно сложно.