Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу

«Криптография»

Группа: М80 – 306Б-18
Студентка: Макаренкова В.М.
Преподаватель: Борисов А. В.
Оценка: _____
Дата: _____

Москва, 2021.

Постановка задачи

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

Вариант 14:

n1=611219701749111463195451937544251195208759452152827846401772765239293765019
13,
n2=166981202821111487603574159347402180221234004474088470134427127019583208585
679731493672560996991989288043247004768446454156726533680678895840262535052207
722153568875423450919653644127144161477230078248529404392103475354920799309387
715301851663504907633271178215986687496281673059795431500200800112337374888657
642932011377010779739631990411714885737361714602715397638982646136481630238419
481808864438911371804085212946840198558441479176256832689600476668930865222709
.

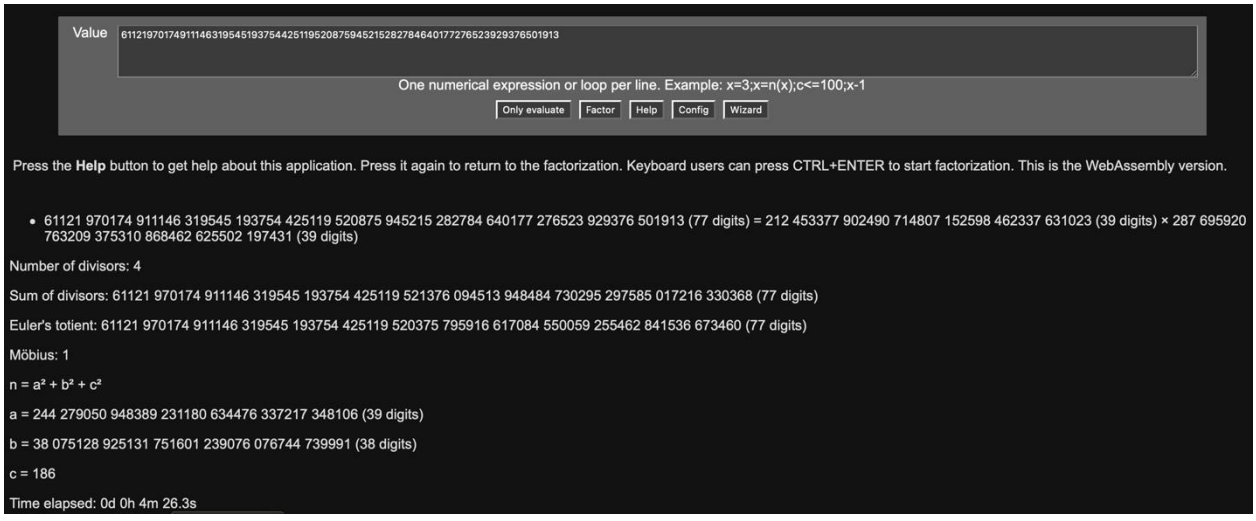
Методы решения

Так как ограничений на методы решения не было, я решила пойти по легкому пути и разложить первое число на сайте 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:

iamverk@MacBook-Air-Mara 1 % python3 lab1.py
n2 =
166981202821111487603574159347402180221234004474088470134427127019583208585679731493672560996991989288043247004768446454156726533680678895840262535052207221535688754234509196536441271441614772300782485294043921034753549207993093871530185166350490763327117821598668749628167305979543150020080011233737488865764293201137701077973963199041171488573736171460271539763898264613648163023841948180886443891137180408521294684019855844147917625683268960047666893086522270 9

885737361714602715397638982646136481630238419481808864438911371804085212946840198558441479176256832689600476668930865222709
from nnumbers is
164002213211582449439345103101010580261572585169552138300989634438730183511532326279004028254034511611436674562580240536376226692152792624551569104138918952496750359796132275664874619889146598902097542761623680838097442518495171412837429189216783713021962416264736667759184361105350899864242921476311929824199

factor1 =
164002213211582449439345103101010580261572585169552138300989634438730183511532326279004028254034511611436674562580240536376226692152792624551569104138918952496750359796132275664874619889146598902097542761623680838097442518495171412837429189216783713021962416264736667759184361105350899864242921476311929824199
factor2 =
10181643256587384320779819782530205592824149924342228406041913425447916239204749909375543893373224862895177713092334026998378017806469703234490024346259491

Код программы

Lab1.py

#SHORT NUMBERS n1

NUMBERS =
[284994967805859272853477327862245466978346919806585432133556769959269315271111,

352358118079150493187099355141629527101749106167997255509619020528333722352217,

119760639583941053725652803731328419697649739176243841021915621242807618608591,

344845228130159226488163571070417679235025139015802019152516926202711846660141,

160769357899975610828199539114109518167531134514190990785144666932076614717841,

274114822339589629024026495441557479713813228028980117869052278950681241194819,

108762353292448487441247663685513658893167646930627178946128889967643172154127,

268887320029090028117214498253204095765884136483366193842361283776500643966781,

123248268911937923199906141216645363665087045422689358104089185316148911496103,

284994967805859272853477327862245466978346919806585432133556769959269315271111,

472379552736871494058143239162622860896965275113543450580272489891667080207763,

361996727456784871855604181056605672088622666207578160811291060873997151708887,

313230894596513941163065516500542159481861849753982064716706926040955753912601,

374456902508739435218273258671224457341348406488533188195528827819627513233269,

611219701749111463195451937544251195208759452152827846401772765239293765019 13,

383456614884902466726252731294544234658015390619372835826246625499154384118189,

242587413455689311805941697582103544343444025737930609728129303011307601823551,

181552877565998943910618543225528579935321447209736978912489118450818545230489,

319373613270896663765954115654922624879359841665992852658124487372881123570003,

3744569025087394352182732586712244573413484064885331881955288278196275132332 69]

#LONG NUMBERS n2

NNUMBERS =
[1342124472692680814864696039831657201341930170537490888948185909193404926961223536479474040666460885376378742819197105748650798865879482304771084236256654384273799702336475969647451700146534430052845833553984165082992842025899656567922774484313656763793347645762888379692094136645059343771155043690219662830401572931468290058814304724439824204259980816710796240835274604462076123461698477384443713751284482994607430755155834233283681253132280989394989471961817143,

1695128485402083763773247025508607781296883851800934596605324477902989989672390098441314233687038522543796524362932674511659084990877094461405769068305253980165481952276151264282270169307424982451349364468884452626363366332792106697498300154504289109043538314722171490851577202002936469515837846884472685701320555954675270470981711883452876152967636160722991943031737727674462234803964546522349706678813412341712703190842025567979822278829254837642753739546649159,

1916242087180680156861712994509728052535159091128844805658679025296716559404434664811725619186652725901325774649017594144788360637407178476936316915220758144535681964371311657071750970414707218112222280453951875213591639735019844579642622014874212594838041457800464921182345127496460888250084171815540351211745813542192969624108567504481905290317359415752535077985931507909722167364312980099834023023021212767107040301344392783417575981002593796696074442689507301,

1598756544210860812002683252504666631284038535154979340910964824673923578639226397918134429192737005854188177977059177858243855990803981275665690912975534091041361701843465578101733863479781680791655959578320442108371634048374313524202193198694894536452471646868825144743014452957912743920239954473534374422647748020165306769379396190044599513110393062461302839244356754741065320775011514774723155863731595182892822790709843296375075272651902641460504103291775361,

1250171497372227982026555999675170108947918951378367343470923483104158597216632066586300921566811265776465427395026458151240042366061271512107752586681699923914902061886213022544496783070727061083763996630816279869169194623169255711135422521925444135939014878277515299870536875962948267973899545621728547726545192382593936985574978881305949487523233148677106330650818223443955800622774189936635106363035784698216185461573761714766211607812695281252356674432444279,

1598756544210860812002683252504666631284038535154979340910964824673923578639226397918134429192737005854188177977059177858243855990803981275665690912975534091041361701843465578101733863479781680791655959578320442108371634048374313524202193198694894536452471646868825144743014452957912743920239954473534374422647748020165306769379396190044599513110393062461302839244356754741065320775011514774723155863731595182892822790709843296375075272651902641460504103291775361,

1611765569148804856242867384258680719850010286298191204635154152942043219729044752688614748313611454546572520541736997794001687127300182565577523301374576898637465463079329544247774787283512154983161737116562645744234565727709746364114005583231547967023025414569413122447328040416970845309432217530722433341506166879058135267652737561086239915598233931006566824074208096468336520404693863268533117447729991162579236036416014409092228354404809885779998800076550137,

14177467869787507650387834432016948376930580071471350079285831921442569467042236590494758980427157782351530260852126352560893481056955596585856196760851613464821804136259107185547729368883111388512812700339059708262004996928275687558408584407339919174540282553261747449656964703936447130918315087871163722894672660845644433050799802860493503622897613938633077951879747971879859575334614760888258163959225587279203300668232112105942963026762617074322173483051121870,

15896869078589605322930419502598074090891160757749059248119283697293085162750729144924473038823436190147828611462774742566344292235738126798298858577225142367897737580736023827542963987467605286204671356869040918576772986866133531605014212545393642155434623305291738232547859578925967439714693310536946287047198975116344940807263844493119113264305436080318461812105908080731040431685156269225193936839179818736338280530681697503531374123421010923268140012860799 31,

14470563577430403187898629612275091047447990814946786123832919869849235193164462877080490779182246565274295436732293643518871833908072627524231172982110419346551522765992254317516715888959815174190264715429324481989444969083616331327076407980393565709505006078950141506587407820420736302617335256351925247739018311504537066619041864399051765841946047321403468580781936233573521469460165494767804910732129539946607701693482114451990193860694698453061853232064399 61,

12624855040201687310008422575815379573283264975224784050024653596488753568102802922445476180707275244175924197767926120587325948529831801486650640588174078660642911795524226275576838868284620610694470321645692350698186694141698828633070326972828021572476527977343920440163200408592574011145240631428946071118295740256009188932533395170616079780684755899312390146830195929916148375233589098062589910776461472469974938947364344953726934440013080012788793957889638 79,

19162420871806801568617129945097280525351590911288448056586790252967165594044346648117256191866527259013257746490175941447883606374071784769363169152207581445356819643713116570717509704147072181122222804539518752135916397350198445796426220148742125948380414578004649211823451274964608882500841718155403512117458135421929696241085675044819052903173594157525350779859315079097221673643129800998340230230212127671070403013443927834175759810025937966960744426895073 01,

19603440006734480101099661237982591387883122230001102854441389846870436820919184377265648736526559593379272139428292838436152529262817891963724717308924224522305311182653859231485873649563920450252677624041195978388744710390172532363083063745412743753556715009911963945245091922784874734290220678484601501149189968384154016448203244939418620612085846868405940252237869240794442627140954903017720771263957902359998360039712906169888947253730020421741485274489917 21,

16884322685356525369761615442254049333529173484668807416465552360809404683693905337775669013748638460889263027167049582533490134650171716874765143454540808295122280915543906952422262227102232713674807533081577925498686812409437301845453047816330110439273275841750941957020629469043067356733549964159070141955505905504722553459616372496410129280190985181933634584156918022437057685037866498824267397680626946780228135270677272788424467599986393125872460984936775 73,

16698120282111148760357415934740218022123400447408847013442712701958320858567973149367256099699198928804324700476844645415672653368067889584026253505220722153568875423450919653644127144161477230078248529404392103475354920799309387153018516635049076332711782159866874962816730597954315002008001123373748886576429320113770107797396319904117148857373617146027153976389826461364816302384194818088644389113718040852129468401985584414791762568326896004766689308652227 09,

14169084447719341143272360643356951750338555687245147232760909092389022494507611631161792983700976377366095987469785396811908061750237394378249497902031141954472876211921620528639113700302812533115824770238590279848186791082392676007634118911135781819389783413687636778555346854134274372902392765730783654373168911955055844636426697161129367283730885533859028643592189337506274405214704767741787934130977543281068768100090834326282132886721944207546209205488511291,

15109385843025147460686876803591387120848268695317498338161525361070299566943782286650144848099932846806364650453365846700065126924820571688588052517305224124355755370476387591838494378611695821743531006167608614420833389111629829780186546090734874555618344725646474341106448770186119465437436805540314573902315148010605642969399036239279990866481377552631038345038332671300460449150826133047599402952702220438132324240801480483055996850135609380612773088576264939,

15033499906313505127942896843130782450400802347937492882843881028115293186513341863145092476540091725800064574393561521328602108813560462716993292012530584330361623216743051882118851116282237496594977166868163840331786137975686189271737528006927952316222354349343355004965993153577865952088162134894290906187247294161317469653368470808158015990205733111105113749510972776073107995920975786223684234407181645957200918994270361355390957408076391671959950085809104331,

15406225094908179490535246491659819823627101385901456801083676605923009421074555721288749853133175961437905691459584340477262214693347263610455103335077100041993455290528251101324297893844389907069275490156858433293934015201542373720790986683248170112968253028251028561435745802110146265300808951070303379347718785548915972867011694369046393627681841310401546991286457558418299989014726795657302590989466425402873370832842632874326160946972589939452327670137815 01,

16265705923840344012310598594084552548100509114311455807738173203854456785977766950683127961452586180126554485218163161080222787625202392679798991846278167936565809063790777458251309334207819198020137034051556960335295557939983593891737558873668573291313432061486325062585463987617255877140830088283477270714347719449643648297679057781869127718984315010760253784801108318403324790207832062061904051003949822187692691563935316046036041428410910392654850704146722591]

import math

v = 14

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

Благодаря данной лабораторной работе я изучила различные материалы о факторизации чисел и алгоритмах факторизации, но все они не работают достаточно быстро на таких числах как, например, n2. Из-за того, что нет какого-то определенного эффективного алгоритма, задача факторизации становится довольно сложной, но в данном задании помогли знания об общем делителе с одним из вариантов.