Московский авиационный институт   
(государственный технический университет)   
  
Факультет прикладной математики   
  
Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу

«Криптография»:

8 факультет, 3 курс

Студент: Стрыгин Д.Д.

Группа: М8О-306Б-19, №25

Преподаватель: Борисов А.В.

**Задача:**

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители. Ниже представлены 20 вариантов. Вариант соответствует номеру по списку группы. Если номер >=20, то взять остаток от деления на 20.

**Вариант 5**

n1=274114822339589629024026495441557479713813228028980117869052278950681241194819,

n2=648537444144074691466518028186818676759368581352429564910762163149201051506031746927275384848791396742709527250142727852431180039190923535159056929623993281020099481411568894488787357221254948793190790806275981934864827272926659859503148740956497215151522596023735212202543224912276297572023521037706733104444718369747900811115015868257500777157415948456055505746368581396798021013597725674053784004359417093528393208322621349799036805557517486465226264479201953

**Теоретическое введение:**

В современной криптографии активно используется алгоритм шифрования RSA. RSA сам по себе не является практически надежным (semantically secured), так как при одних и тех же значениях входных параметров (ключа и сообщения) выдаёт одинаковый результат. Однако, его удобно использовать как вспомогательный алгоритм, для шфровки, например, сеансового ключа (используется в TLS, а так же в ранних версиях PGP). Несложно показать, что задача дешифровки сообщения или ключа, зашифрованного с помощью RSA сводится к проблеме факторизации целых чисел.

В математике факториза́ция — это декомпозиция объекта (например, числа, полинома или матрицы) в произведение других объектов, или факторов, которые, будучи перемноженными, дают исходный объект. Например, число 15 факторизуется на простые числа 3 и 5, а полином x2 − 4 факторизуется на (x − 2)(x + 2). В результате факторизации во всех случаях получается произведение более простых объектов, чем исходный.

**Описание работы:**

В начале была придумана идея использовать алгоритм поиска простых чисел в заданном диапазоне и с проверкой чисел на делимость. Первоначально был реализован наивный алгоритм через цикл перебора делителей для каждого числа, после решетом Эратосфена, но в конечном итоге была реализована утилита msieve, созданная на основе **Общий метод решета числового поля**.

Второе число не поддавалась и для него. В следствии чего было придумано использовать дополнительную информацию об НОД среди имеющихся чисел.

В результате поиска был найден НОД и второе число было разложено.

**Листинг программы:**

import math

nums = [

    9856374462285180827430882504693482921047255832047915840153891370083550094688187,

    5684417577210707125270927756395826164432807313246637831293635062503262393683373,

    11094321979214883522081243833073971863144077429544717183070098194022031113245733,

    16070673494387192010779398962456898719302485586995885901671443447466287976435079,

    11523502327590549981219831269946841237309424301758719772739686792366969245994201,

    274114822339589629024026495441557479713813228028980117869052278950681241194819,

    108762353292448487441247663685513658893167646930627178946128889967643172154127,

    268887320029090028117214498253204095765884136483366193842361283776500643966781,

    123248268911937923199906141216645363665087045422689358104089185316148911496103,

    284994967805859272853477327862245466978346919806585432133556769959269315271111,

    472379552736871494058143239162622860896965275113543450580272489891667080207763,

    361996727456784871855604181056605672088622666207578160811291060873997151708887,

    313230894596513941163065516500542159481861849753982064716706926040955753912601,

    374456902508739435218273258671224457341348406488533188195528827819627513233269,

    61121970174911146319545193754425119520875945215282784640177276523929376501913,

    383456614884902466726252731294544234658015390619372835826246625499154384118189,

    242587413455689311805941697582103544343444025737930609728129303011307601823551,

    181552877565998943910618543225528579935321447209736978912489118450818545230489,

    17024488909070023298334718577391876794173841440968819508442341693309582028164211,

    6774663650526306188279960092002181753216588352015040398903835496234293805285631

]

long\_nums = [

    5401357812801580025919761371182225752432497493775184794697572547224195271992571426283909590106860788464786654900893304329348861804228870690569171015755809935445924265497255498176358044672917400832201143434137825294420722962135913707142334254775857657776485041271833454580492262250586297878059894897967270599446437536164564824226824084414404901981300802805483250936961401767891815086247808680628898041247011403210366263005799418053706019834932605092598030696547627,

    3302022959000306046128783870517426861536127416885126746405163395921117861821121029527442053342211074722025278666410267389648339529546822960355263493733516389988478563166173570125125320373840042467197427715570566783354934876492962376348888456514955245307546351321990013280137573736294484549777438046907147746082497547574721415593280178417596118368966005440070935516884807614630932600250786098413254455580028740515858031572232760606988105994915916825321411634327591,

    5642491146411201705731808942454930447273303978490515512906223745654947807393552900639171213484932874743249654507714130436835145752483146038413764971943257386263828340471807133765288577578312521737870023393049964811784284205858662738446457057167022507432183807033334643308741690468418067924532705024708709945308011199701967610097050560558138842484332849220413849278724541613669597680013960381261461013731123869489074067695113937598392668449297686984423553477740729,

    6842150046087882095119252943205809109872498280385038962636286718548788348679906761214254719009387218303777984125116167531808882261705659130142474352277906725407290120015950526800477719814158545556444377490334842429415721687383380060868396734727006024348680651384672606344662788501245208450457517675610171962879688760721754166650519379394090393387441824339439135781371859966433884401412117922290338042975984057736333004792080731559120439155480122523672807955325817,

    6510659995266063591055365197843971653052867831776645052653874807327235023002495154291390621749600384259832142220720545046622676463608117082696608705726376169500650532288828439750578807097406651869250241559458475603776185255754047995821599563606418243192330983996400237750221371151693489272662706514158594056006246728824717597157510703698451575550535588715006050673543204274350358917225775526580639305331350371746689342288675972636204132354942034816575946642646647,

    6485374441440746914665180281868186767593685813524295649107621631492010515060317456927275384848791396742709527250142727852431180039190923535159056929623993281020099481411568894488787357221254948793190790806275981934864827272926659859503148740956497215151522596023735212202543224912276297572023521037706733104444718369747900811115015868257500777157415948456055505746368581396798021013597725674053784004359417093528393208322621349799036805557517486465226264479201953,

    6460022354312582572793343100604163087216372941214865569096184912826464912951934844843290548057971769185825571008898779827285829198579888230136431509729644521505820664210671796785408727881787088472858544237323843649456753195786360153338730014417334774860737788349377171270117783539517147103729867405974761065566886813619687414228558901198237125901551456589579117850298162443488215653711057874251971726449089251928969082516345596779536154854135917899503811275677859,

    6290786896526191101104637204956784381227340661051657910828669971626733569324602870777401334764597738095098477634387696990987529424753021134530017265150211509517870601779076845275000366130643050644003846177805527625055452287543233874755833265955939691167304717135574631233249922210712198606275695435254964218983231064188142002836548716474270002795941557533485168611316063143230252477892493843858478050872063688267933199443582315041224037924767099733678635301638141,

    3968620073611058415100520213411260157904795081130719522978235660633005593445914914223796459264398850320575005838313874062507797504699598472587275240219618752357750196438182065276333348805093637278937115851245038648988552443219178175055389585371559294805102865430117693527691196682205767860489009162383359596893303280433460505286876560700748190262806050526438707955489719706049045117892865324610657207096287410543113618086241243889950193461784940722479757469151539,

    2863197031529473288153933699822451082727702000722696759885229504131399907282349485940086747522374189223077543691662341751318826771131086228894563121717687896960912413973444974266919326734961535816370149394920104197026110496592400216298456410229697005431188357441665617874254007522061912001851031115817424233075402804171424247915010100050176832284553805429250833518052092163433548366573200194191867920554465873387553772422573481328508708279162428700310652039850453,

    5742406534529289317346806810998293126135961121255923767806998078366919906346395893687518748524572892254259619494248139370066046583738624380215695015213890974422431279510618672126213957060039336103893369697640285320473263985513279456769716303734724009078244726116304929241148439704054781477268492894665994909866506431647367801547667275806259100936872459426615298576113470047343542208737928808413922671034766241809092439197356071672909937228180990077950407664219007,

    8390215390741416296165728438444990265873752901278087636077774776192134148075025976941079883829710135873444808375678749989841386035006794753177002125910135642536563166093404194081384100200527334183948439259777254016779633390822424664157982455073549929911569591005069347891904430796714001619918137197111765829747653465144034964302618098044927773419518450730198825120729168503166440363846084663171050507797381039040530910389182301934645605548291059568523203447442799,

    4399718557344668512982825468593339414449818136881809741813232714125048548028800001295290189077224755408700120793204788243839645884033454205632195279782108412849274735824506406236572391388536172362446436730597596869126652904206039856863066288725309001561486825308360428571243719562053157988491756432402025968521057920213915133759226444843156745663470177133420679614125824445862395006301312165904622674862571282606855115068547823738210731074919273694772322590742003,

    6388532302085669228615771388983452948007941743568163970560946831157382344058239126229806052859737888487783937846720963023038153065348279670854149017787470574812006836885115106891910821052589226052685423814611520013606076441149759993153875258431039836687849147729558354065212066523592281254609943962306220641867429175693807391177713424759495700540370707278314153079064977575088510104373434672731111688909374407992653018176506344174461412097021874839013850305001081,

    6238596931990131478275327152343801799668257762705582576427997815976220309207691211435205004903767290173230872141307213029392296401324769866348679434743265963571375902347394343714110590043962617817326709729518180348452284402707055765832648925000626213596053865173116238603592051986073329528908502125295911837124501151097345605082152827287279961240150766589609665614675527012229363982035766005443366925574863537569235988977028475559932462781742771732084527629722071,

    9173108187535281517140762116700384612326624554619159756170327131075765663592428184342498108166844987875472192379425402617261537751017461789175811281066290112144995485771928277678645046268515605836413638915408097220188140275180089310734305255138886443749996612234117011911904572687273790898184986984786012309336821986211719508683069979735549322015707035016397961277189356172028200502143324154428183926213506337495841035478668065439542819480000104949666864308342553,

    5689251455873233118490724339174324094087448014557865887166027049566432196203144154436175954958724212513309242872121824618285332514583702419554960431547751766766929616570699170194830343448484017300552001636746608091709124028460097687944743670363346860455317135997371448237415568085697054394924321902983517319757271201684201741435608536263437954474805803792414559809251069498675116829203600404115287783073317258677431432165073466071636945594328043962958101936147809,

    3296598709197109112959820504704293475825900372427606827655635749866085096535606800975862283061728734524969268799233153479808607126273973940959311490683805332230225342213747295106136823594960061876340564055040323692050623065242546198395644412628393882345454746008952281334548492319865710133232878730027509283080732639248652119682150346928729193906149854800998865336163879389183767541563672832357406596471532424507051233164360205118091594498781416503536663813499217,

    4176791439387440012105039841955024735745096093311965085776705389569193465736477338711973534061613622598861652925070295618921160489665913657434787707382436989648017257052900453045172383713914470046749292919938264872421222262899730111739865042752534532703648831594593208526774090647825092475529740690380050365466740698687728311258060940910216674003202106179220625882192773114972092326972470824029802880077228618432257053517880272289765956917405478998570309852861051,

    3809251286382179803016660472471937683846303761815393532574144834657645774450997469367539749723753364997606495892557318308598177540157938785987841533102654275745244670142781673547369698875645196279215803808112355168274026152117039595642690835965916238959137282786396037992485956499183848286816537213097381861822182688494084368434511461011178225226924881381882148176080895824110153575203695988826769890236133923107190668175242183002350816068120098977871980086991879

]

n = long\_nums[5]

factors = []

# циклы поиска НОД среди имеющихся чисел

for num in nums:

    gcd = math.gcd(n, num)

    if gcd != 1 and gcd != n:

        factors.append(gcd)

for bignum in long\_nums:

    gcd = math.gcd(n, bignum)

    if gcd != 1 and gcd != n:

        factors.append(gcd)

if bool(factors) is True:

    for factor in factors:

        print("factor1:", factor)

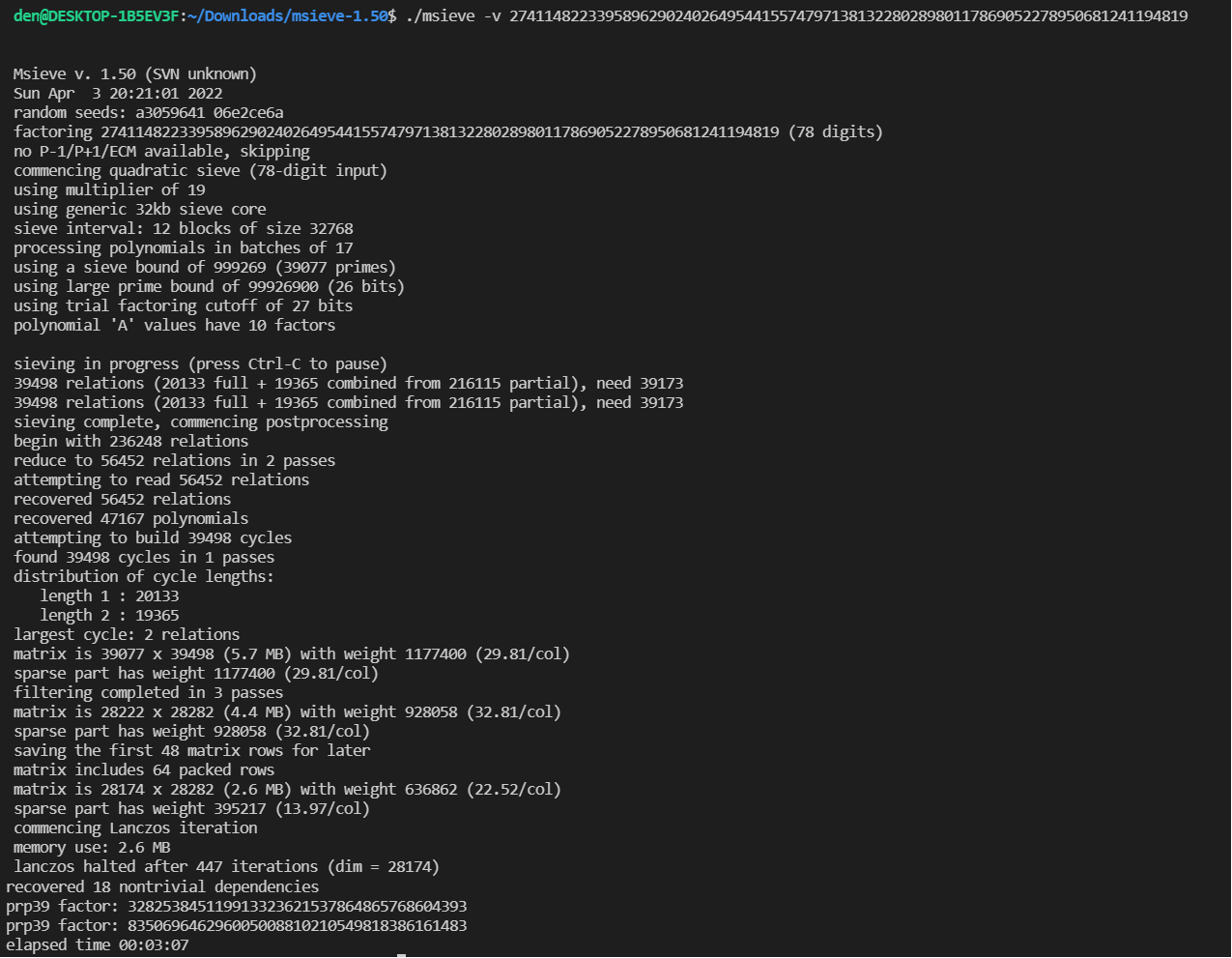
        print("factor2:", n // factor)

else:

    print("Общих делителей не найдено")

**Итоговые результаты:**

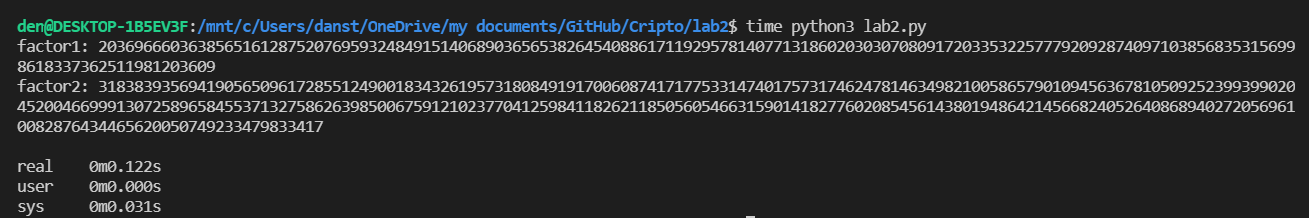
Первое число n1:



n1 = 328253845119913323621537864865768604393 \*

835069646296005008810210549818386161483

Второе число n2:

****

n2 = 20369666036385651612875207695932484915140689036565382645408861711929578140771318602030307080917203353225777920928740971038568353156998618337362511981203609 \*

318383935694190565096172855124900183432619573180849191700608741717753314740175731746247814634982100586579010945636781050925239939902045200466999130725896584553713275862639850067591210237704125984118262118505605466315901418277602085456143801948642145668240526408689402720569610082876434465620050749233479833417

**Трудности, встреченные в процессе:**

Без подсказки про использование уже имеющихся чисел никогда бы не догадался, как разложить на множители второе число

**Выводы:**

Существует немалое количество алгоритмов факторизации, которые в свою очередь используются в алгоритмах криптографии.

**Литература:**

* Электронный ресурс «Хабр» URL:
* <https://habr.com/ru/sandbox/163811/>
* Электронный ресурс «Википедия» URL: <https://ru.wikipedia.org/wiki/Общий_метод_решета_числового_поля>