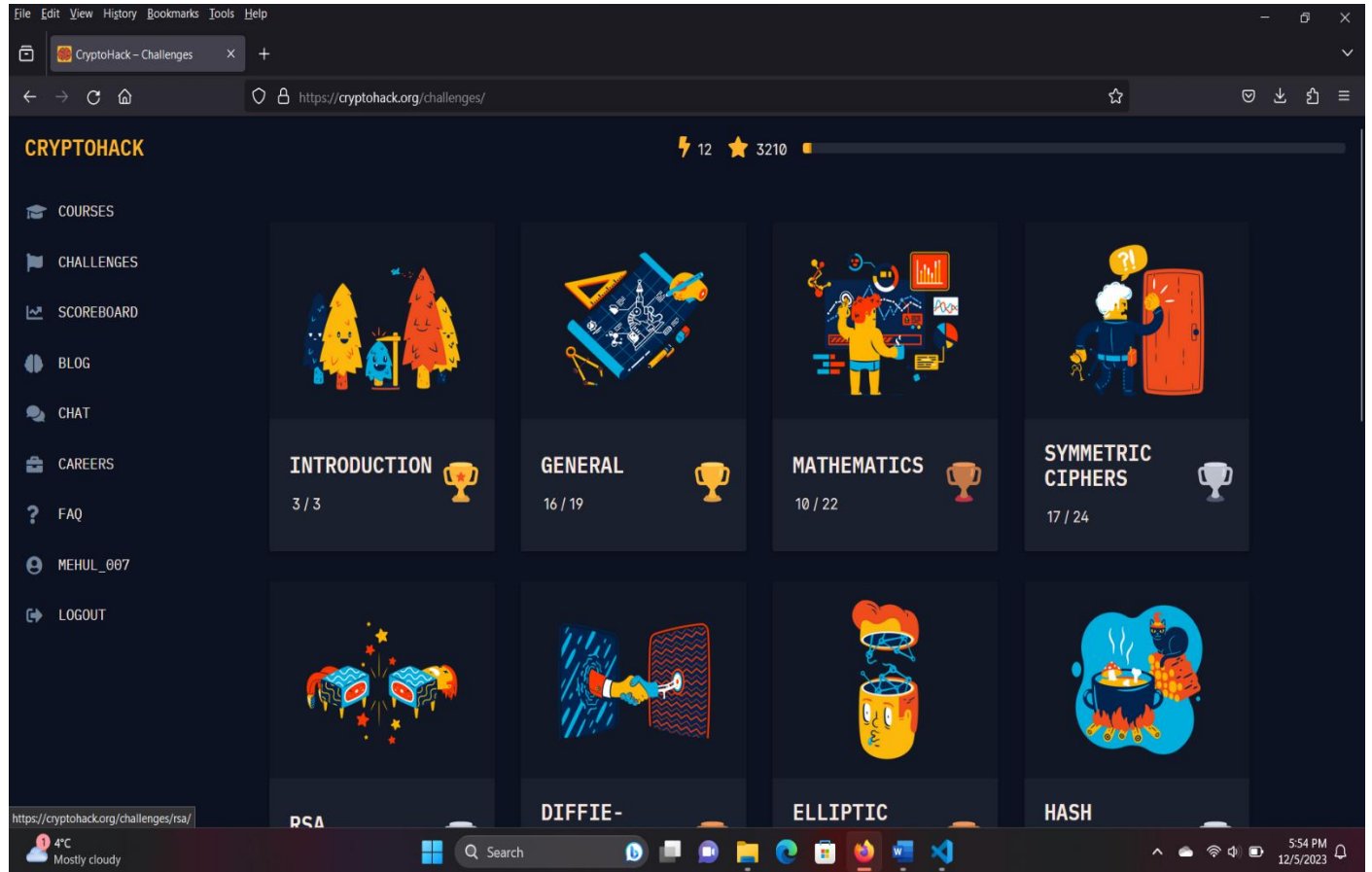


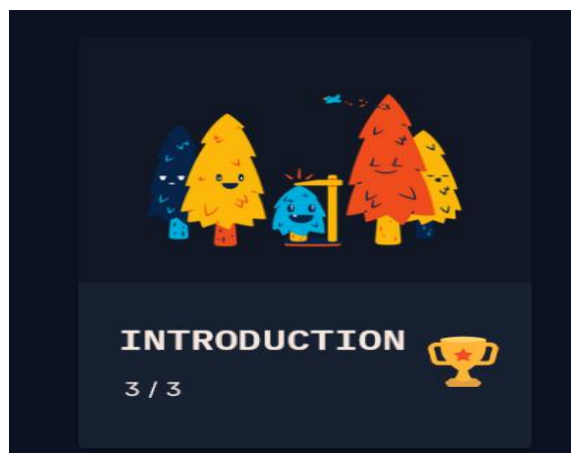
# Cryptohack solutions:

SCORE:3210

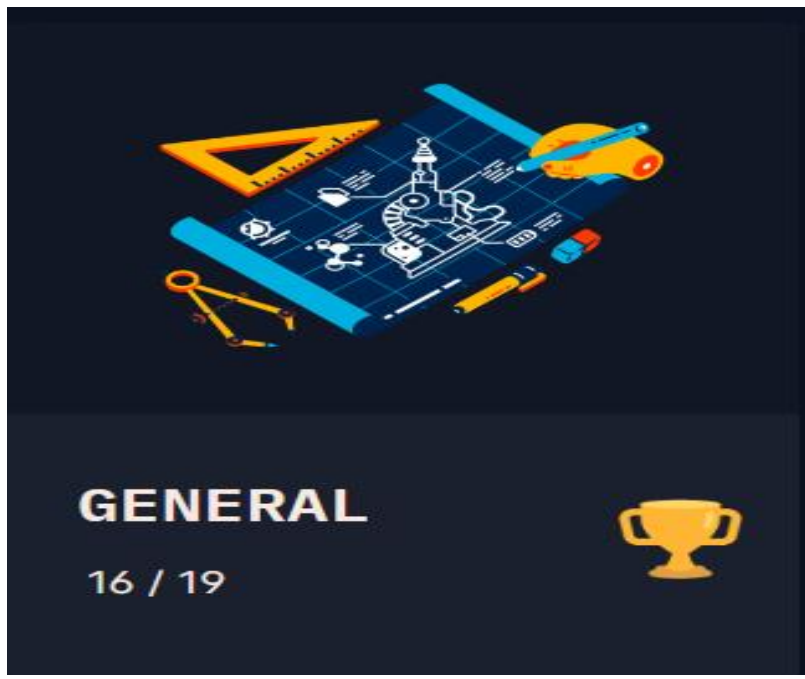
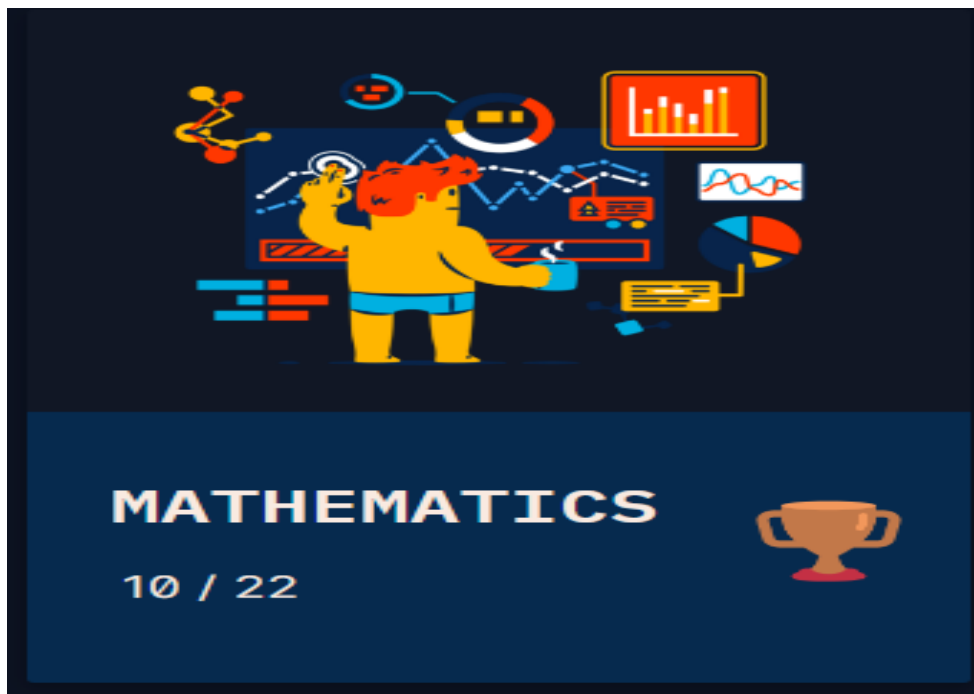


Challenges:

Introduction: Completed 3/3 challenges:



General : Completed 16/19 challenges:

Mathematics: 10/22

## Symmetric Ciphers: 17/24



### **Code For Crophack Solutions:**

#### Crossed wires:

e = 0x10001

n =

21711308225346315542706844618441565741046498277716979943478360598053144971379956916  
57537034344898860190585457202963584662625948729795030523166110985585494749420913520  
55892586435179615215949243684986720642932082308024410773901936829580951119220826778  
13175804775628884377724377647428385841831277059274172982280545237765559969228707506  
85756121526849102409706392033772178367306053018163716157740158912655855618254689678  
33073705172750465227040473857861114894470647942100108027617086159072455234925858962  
86374996088089317826162798278528296206977900274431829829206103227171839270887476436  
899494428371323874689055690729986771

d =

27344116772511480307231380057161097338388665453755276020182551593196310266531907836  
70493107936401603981429171880504360560494771017246468702902647370954220312452541342  
85874759057627377510787045085353371711668432697626300643573338204580797189076201874  
77295740210574303317780339823591848381597473312365385018499653292647749276075704103  
47019418407451937875684373454982306923178403161216817237890962651214718831954215200  
63765110390720934790085782472265321717954814814568718137722054486452180823012273096

```
74529814353553349321042654880757776386080413252567762752000675415330225279647434785
54948792578057708522350812154888097
```

```
c =
```

```
20304610279578186738172766224224793119885071262464464448863461184092225736054747976
98517967390544150268912621628289770450874540379905473412158396885399979160428161515
41007362591314534243853643246302296711853437781728072626407093018382748246031016924
85662726226902121105591137437331463201881264245562214012160875177167442010952439360
62339665897441390046909383679475227039952007459632905872587483408218869737759794940
57790391391941960653644262132083454614070307710897875292000571057465844935547227905
92530472869581310117300343461207750821737840042745530876391793484035024644475535353
227851321505537398888106855012746117
```

```
print(bytes.fromhex(hex(pow(c, pow(106979*108533*69557*97117*103231, -1, e*d-1),
n))[2:]).decode())
```

```
Flag:crypto{3ncrypt_y0ur_s3cr3t_w1th_y0ur_fr1end5_publ1c_k3y}
```

## Infinite decent:

```
n =
```

```
38334771233087704045223861932952484176339252614684057223292692464209489145397924638
37989133941143053683604268670216236496670242172665290008597035425903160633185923919
2506201422967142377796679798747131250552455356061834719512365575593221216339005132
46433884719524862763962348712402589069341630578816090576201182507933688056746103332
2240015771102929696350161937950387427696385850443727779964835844646100463807227367
90790188061964311222153985614287276995741553706506834906746892708903948496564047090
01430748405460986212953026210866956783472635207806008188971210941207373102603046630
00603417375042238220147140564137521658417493681595105881786040961919567509410783914
15634472219765129561622344109769892244712668402761549412177892054051266761597330660
54570431721056775982875715690477849560896878574799805985746744012815606839174691968
42582276828660836623452636595580668641092124572861145062284709307750927353853883162
68663664139056183180238043386636254075940621543717531670995823417070666005930452836
38981212946205177164604849839719515740538692344689388659304868098489698980913580227
68929110385880087019267292698124532268917765460376635838936254792526430425171969589
90266376741676514631089466493864064316127648074609662749196545969926051
```

```
e = 65537
```

```
c =
```

```
98280456757136766244944891987028935843441533415613592591358482906016439563076150526
11636984221310333348050670599363390199410728189018724849550727086862138465220769760
70198991664921324083487892525551964286086613206718774127104897823582820113641277995
63335562917707783563681920786994453004763755404510541574502176243896756839917991848
42809159491911144802394852776636830450310065037991415305819114007252809589857601889
38298301043621249271405551079941141430422667587093280689026640378700757425421943180
59191313468675939426810988239079424823495317464035252325521917592045198152643533223
```

01595270264924949475339510097353454176628555189185964932037117856220025222877939539  
39741697369985233945985171741821420074805266030255780046659368546572945413386975135  
21007818552254811797566860763442604365744596444735991732790926343720102293453429936  
73420624610996881715881574992706356183527463619514970231741568040198715033699458375  
20625652376059531537903711559184399411934014732717530381805601297841928003516497244  
65553733201451581525173536731674524145027931923204961274369826379325051601238308635  
19254022348405509620329340041981602411179790344286418196595924774500682269096792095  
7905188441550106930799896292835287867403979631824085790047851383294389

p =

19579267410474709598749314750954211170621862561006233612440352022286786882372619130  
07163982410978354056451242908167413233681197240456395702546503402578120646663173078  
45163372102913343563964717321687427397904641098810392194525044566115891543494273038  
32789968502204300316585544080003423669120186095188478480761108168299370326928127888  
78681939237247706951531817975170298580902421016424340954469270868421504222693208105  
28310285700603089630932176221831116433356923610198974492654022905400257905815899808  
67847884281862216603571536255382298035337865885153328169634178323279004749915197270  
120323340416965014136429743252761521

q =

19579267410474709598749314750954211170621862561006233612440352022286786882372619130  
07163982410978354056451242908167413233681197240456395702546503402578120646663173078  
45163372102913343563964717321687427397904641098810392194525044566115891543494273038  
32789968502204300316585544080003423669120186095188478480761108168299370326928127888  
78681939237247706951531817975170298580902421016424340954469270868421504222693208105  
28310285700603089630932176221831116433356923626352035828685261788380189469867926568  
19885261069890315500550802303622551029821058459163702751893798676443415681144429096  
989664473705850619792495553724950931

```
print(bytes.fromhex(hex(pow(c, pow(e, -1, (p-1)*(q-1)), n))[2:]).decode())
```

Flag:crypto{f3rm47\_w45\_4\_g3n1u5}

## RSA backdoor viability:

n =

70987244318676158212574758566872450126855845855879867301467348376630096483647916724  
13156600538786504217617266398720898855020049024874719464109184209276825863621111373  
64814638033425428214041019139158018673749256694555341525164012369589067354955298579  
13173546679591852281612739834046576140671906028409809464328939001631166831668780883  
75635891240918677736550449130036685909548997053667870809237172708271842226737068561  
84434629431186284270269532605221507485774898673802583974291853116198037970076073697  
22504709890141463743339265850067074099600879986053003251571603144978708937140348520  
58107958804169206421864510223749898916119439068911390477640420510716472030575201042  
67427832746020858026150611650447823314079076243582616371718150121483335889885277291  
31283408323408766039953466583529162123205647384322451590902312083437766450578832952

75179321609090134109333125728102080438495296552094200551806807757186140885210147724  
91776654380478948591063486615023605584483338460667397264724871221133652955371027085  
80422395610453260411396911971648514242499625573737646483431552782256601792359862663  
44380667247635599434410235745751689240102742613768632025983534300108751829474851010  
76308406061724505065886990350185188453776162319552566614214624361251463

c =

60848461731613812644327566052426302550813538374566517543322959851743303000370426165  
81725823705437582776855475338340858995410361565954892063692797392109041547164645956  
57421948607569920498815631503197235702333017824993576326860166652845334617579798536  
44206618495355097548703172108510575766780083817222594700122449512639058795034682297  
85196776735681215954278279801953324647470315774319259373142093914334076848457971711  
87006586455012364702160988147108989822392986966689057906884691499234298351003666019  
95752873809433038977505448573144827459533032297688687552852522933751290995239104128  
00064260033007205477210727251685001046519619702927713823906477514504458923613113320  
74663895375544959193148114635476827855327421812307562742481487812965210406231507524  
83088937541904554205785867960926538986933233181121860144037312179746131893197689067  
43368075281071154239151527092652375903583483487165436839000846409214757972663904553  
66908727400038393697480363793285799860812451995497444221674390372255599514578194487  
52388203823448787222354051300473403913524384955131506529773753511252544009417139303  
96229925615191708499628916451961113075373411946216897972824962813022970260251317434  
23205544193536699103338587843100187637572006174858230467771942700918388

e = 65537

p =

20365029276121374486239093637518056591173153560816088704974934225137631026021006278  
72817226306709337512779951702164268302645394189208554959641555963283714007258774330  
55744792186283881915870602622631704303157618903039902338715768605511661621105655750  
88243122411840875491614571931769789173216896527668318434571140231043841883246745997  
47450017667192615361616877915240030631336247788826299709303613658231888163323537602  
62764168296528852232344113391163627325903147313917709424336259927104753940216755725  
75027445852371400736509772725581130537614203735350104770971283827769016324589620678  
432160581245381480093375303381611323

q =

34857423162121791604235470898471761566115159084585269586007822559458774716277164882  
51035886947629393917628761027489950978673682446174060361859854994527302947982529045  
90623704246574461516239056536321816780659754729682428228599269024630437306449584679  
21837687772906975274812905594211460094944271575698004920372905721798856429806040099  
6988314717097740990034411156884344945240754279932746794468563025874802887510344476  
01525874935437991856466926840324608581509607904955759214554231857098113426891851279  
36111993248778962219413451258545863084403721135633428491046474540472029592613134125  
767864006495572504245538373207974181

```
print(bytes.fromhex(hex(pow(c, pow(e, -1, (p-1)*(q-1)), n))[2:]).decode())
```

Flag: crypto{I\_want\_to\_Break\_Square-free\_4p-1}

Everything is big:

n =

0x8da7d2ec7bf9b322a539afb9962d4d2eb3e3d449d709b80a51dc680a14c87ffa863edfc7b5a2a542a0f  
a610febe2d967b58ae714c46a6eccb44cd5c90d1cf5e271224aa3367e5a13305f2744e2e56059b17bf520c  
95d521d34fdad3b0c12e7821a3169aa900c711e6923ca1a26c71fc5ac8a9ff8c878164e2434c724b68b508a  
030f86211c1307b6f90c0cd489a27fdc5e6190f6193447e0441a49edde165cf6074994ea260a21ea1fc7e2df  
b038df437f02b9ddb7b5244a9620c8eca858865e83bab3413135e76a54ee718f4e431c29d3cb6e353a75d  
74f831bed2cc7bdce553f25b617b3bdd9ef901e249e43545c91b0cd8798b27804d61926e317a2b745

e =

0x86d357db4e1b60a2e9f9f25e2db15204c820b6e8d8d04d29db168c890bc8a6c1e31b9316c9680174e12  
8515a00256b775a1a8cca9c6936f1b4c2298c03032cda4dd8eca1145828d31466bf56bfcf0c6a8b4a1b2fb  
27de7a57fae7430048d7590734b2f05b6443ad60d89606802409d2fa4c6767ad42bffaef01a8ef136441836  
2e133fa7b2770af64a68ad50ad8d2bd5cebb99ceb13368fb31a6e7503e753f8638e21a96af1b6498c18578  
ba89b98d70fa482ad137d28fe701b4b77baa25d5e84c81b26ee9bddf8cbb51a071c60dd57714de379cd4b  
c14932809ba18524a0a18e4133665cfc46e2c4fcfbcb28e0a0957e5513a7307c422b87a6182d0b6a074b4d

c =

0x6a2f2e401a54eeb5dab1e6d5d80e92a6ca189049e22844c825012b8f0578f95b269b19644c7c8af3d544  
840d380ed75fdf86844aa8976622fa0501eaec0e5a1a5ab09d3d1037e55501c4e270060470c9f4019ced6c  
4e67673843daf2fd71c64f3dd8939ae322f2b79d283b3382052d076ebe9bb50b0042f1f7dd7beadf0f56869  
26ade9fc8370283ead781a21896e7a878d99e77c3bb1f470401062c0e0327fd85da1cf12901635f1df310e8  
f8c7d87aff5a01dbbecd739cd8f36462060d0eb237af8d613e2d9cebb67d612bcfc353ef2cd44b7ac85e471  
287eb04ae9b388b66ea8eb32429ae96dba5da8206894fa8c58a7440a127fceb5717a2eaa3c29f25f7

p =

11550729043680468185397251378585522909233408035687471788343423823553266444140069832  
96427512646522995762986365630345661549368128949762008111163956276428241298812018796  
64681775402664283913508399125714656956248098339209538838857780042711388040763913184  
276611985732635893909527514214876581238108724839614805837919

q =

15481583883073575626683989700213231453867590913524995485254210017959019205525710060  
17595234014093800494632662653237804065546922480122244854694683553256989798250727263  
57567352976501667578692853621821116743497998613306224857520639271685772211046709014  
876104481766909036522349223506326852943709940047836080845531

print(bytes.fromhex(hex(pow(c, pow(e, -1, (p-1)\*(q-1)), n))[2:]).decode())

crypto{s0m3th1ng5\_c4n\_b3\_t00\_b1g}

## Everything is still big:

n =

```
0x665166804cd78e8197073f65f58bca14e019982245fcc7cad74535e948a4e0258b2e919bf3720968a00e5240c5e1d6b8831d8fec300d969fcc6c6c11dde826d3fbe0837194f2dc64194c78379440671563c6c75267f0286d779e6d91d3e9037c642a860a894d8c45b7ed564d341501cedf260d3019234f2964ccc6c56b6de8a4f66667e9672a03f6c29d95100cdf5cb363d66f2131823a953621680300ab3a2eb51c12999b6d4249dde499055584925399f3a8c7a4a5a21f095878e80bbc772f785d2cbf70a87c6b854eb566e1e1beb7d4ac6eb46023b3dc7fdf34529a40f5fc5797f9c15c54ed4cb018c072168e9c30ca3602e00ea4047d2e5686c6eb37b9
```

e =

```
0x2c998e57bc651fe4807443dbb3e794711ca22b473d7792a64b7a326538dc528a17c79c72e425bf29937e47b2d6f6330ee5c13bfd8564b50e49132d47befd0ee2e85f4bfe2c9452d62ef838d487c099b3d7c80f14e362b3d97ca4774f1e4e851d38a4a834b077ded3d40cd20ddc45d57581beaa7b4d299da9dec8a1f361c808637238fa368e07c7d08f5654c7b2f8a90d47857e9b9c0a81a46769f6307d5a4442707afb017959d9a681fa1dc8d97565e55f02df34b04a3d0a0bf98b7798d7084db4b3f6696fa139f83ada3dc70d0b4c57bf49f530dec938096071f9c4498fdef9641dfbfe516c985b27d1748cc6ce1a4beb1381fb165a3d14f61032e0f76f095d
```

c =

```
0x503d5dd3bf3d76918b868c0789c81b4a384184ddadef81142eabdcdb78656632e54c9cb22ac2c41178607aa41adebdf89cd24ec1876365994f54f2b8fc492636b59382eb5094c46b5818cf8d9b42aed7e8051d7ca1537202d20ef945876e94f502e048ad71c7ad89200341f8071dc73c2cc1c7688494cad0110fca4854ee6a1ba999005a650062a5d55063693e8b018b08c4591946a3fc961dae2ba0c046f0848fbe5206d56767aae8812d55ee9decc1587cf5905887846cd3ecc6fc069e40d36b29ee48229c0c79eceab9a95b11d15421b8585a2576a63b9f09c56a4ca1729680410da237ac5b05850604e2af1f4ede9cf3928cbb3193a159e64482928b585ac
```

p =

```
98444549679044409506244239144443867459824227934526036052949278261505813439015297459200379108752444235232667213138464076415095486907288282630595622287237215801470940146886371515679909322090871473412384894540642399950010296214525469622505798526072170187467562765920044646574445427364231529083610955760228212701
```

q =

```
131205304707717699800023219057082007986286045823683571663112014612188606710079038751853416273709729039622908861933527111469616900188875912430487264576215232569029320804579614330240773622645122871884209068761138439268551367198798009790636662892148063583135747945604771740458352899202428704645256790931460695949
```

```
print(bytes.fromhex(hex(pow(c, pow(e, -1, (p-1)*(q-1)), n))[2:]).decode())
```

```
crypto{bon3h5_4tt4ck_i5_sr0ng3r_th4n_w13n3r5}
```



## Marins secret:

n =

65841627483018454412502751992144351578988826415607473309924404012621368249771403279  
81163992881765024628292557845259777229030187144343096981082083886647682627543164262  
20651576623731617882923164117579624827261244506084274371250277849351631679441171018  
41801849803999647254989315057718930287152031171517973071431218145624509784849166979  
59972898306129880585239683848088228283709001984892492433991651252192447537907797644  
66236965135793576516193213175061401667388622228362042717054014679032953441034021506  
85601708106261757235119541850589938871570979599202955904211978342359732470710069406  
46759092387175730587641188932251116027038380806185654011399021430699011171742042528  
71948846864436771808616432457102844534843857198735242005309073939051433790946726672  
23464325934953518626857162907793759783880133797309228560874420995153319986822804000  
44321325970733903633578923799976558788576963348922163450702276467498513812085540449  
40444182864026513709449823489593439017366358869648168238735087593808344484365136284  
21972523381160533181500742458289082188726068288663254361310925286211432637207778536  
92925709005948144810974437812695626473036714288957642240844022596051096003630989500  
91998891375812839523613295667253813978434879172781217285652895469194181218343078754  
50169474659873821524376974795657255598959459818063909834489117587945599465238213703  
8240166358066403475457

e = 65537

c =

40028046308893043231928035911519497758251736361053246429521066953040787075343912745  
54013845697054256214459439929633809830849173854286312230469088378041263993458752529  
17090184158440305503817193246288672986488987883177380307377025079266030262650932575  
20514185341330255846036424235553127296748140941478363455879117582781654076754594453  
42381890790301928432885969349796935179646556615073467297519879281470216201650099650  
51933278913952899114253301044747587310830419190623282578931589587504555005361571572  
56191686606345881296531447416049906752506749514015009211962092836300746739092013071  
75211691051679633641546364720550840125921385703543902467792760031561846762987107465  
83104700516466091034510765027167956117869051938116457370384737440965109619578227422  
04980656606057183101761087707248426272478957107652958642740578012109654694281232232  
48071451370179422668635349890821151890655600118411509083809373543012431532064288963  
20576609904361937035263985348984794208198892615898907005955403529470847124269512316  
19175395020379457865602932450668829344657159850604219821908032574732863623204093676  
17885584215289602798328021275621158523049468676283165029595622744854838674817311493  
38209009753229463924855930103271197831370982488703456463385914801246828662212622006  
94738011554952982019735573852532988523217021575758568548440234443789498155517912928  
7164971002033759724456

p = 2\*\*2203-1

q = 2\*\*2281-1

print(bytes.fromhex(hex(pow(c, pow(e, -1, (p-1)\*(q-1)), n))[2:]).decode())

```
crypto{Th3se_Pr1m3s_4r3_t00_r4r3}
```

## Unencryptable:

N =

```
0x7fe8cafec59886e9318830f33747cafd200588406e7c42741859e15994ab62410438991ab5d9fc94f3862
19e3c27d6ffc73754f791e7b2c565611f8fe5054dd132b8c4f3eadcf1180cd8f2a3cc756b06996f2d5b67c39
0adcba9d444697b13d12b2badfc3c7d5459df16a047ca25f4d18570cd6fa727aed46394576cfdb56b41
```

e = 0x10001

c =

```
0x5233da71cc1dc1c5f21039f51eb51c80657e1af217d563aa25a8104a4e84a42379040ecdffd5afa191156
ccb40b6f188f4ad96c58922428c4c0bc17fd5384456853e139afde40c3f95988879629297f48d0efa6b3357
16a4c24bfee36f714d34a4e810a9689e93a0af8502528844ae578100b0188a2790518c695c095c9d677b
```

p =

```
82398353972085161117203628479494254010456723658299376021174804493166945582266222001
10057535873802132963548914201468383545676262090246827792522994758916609
```

q =

```
10900824353334471830007307529937357926160386461967884446160315218630687793341471079
170750548554707926611542019859296605188535413447791710067186432371970369
```

d = pow(e, -1, (p-1)\*(q-1))

```
print(bytes.fromhex(hex(pow(c, d, N))[2:]).decode())
```

```
crypto{R3m3mb3r!_F1x3d_P0iNts_aR3_s3crE7s_t00}
```

```
from math import sqrt
```

```
# Hàm nhân vector
```

```
def dot_product(v1, v2):
```

```
    return sum(a*b for a, b in zip(v1, v2)) # tương đương với a in v1 * b in v2
```

```
# Hàm tính norm ~ 2
```

```
def vector_norm(v):
```

```
    return sqrt(dot_product(v, v))
```

```
vectors = [[4, 1, 3, -1], [2, 1, -3, 4], [1, 0, -2, 7], [6, 2, 9, -5]]
```

## # Gram smith

```
def gram_smith(vectors):  
    u = []  
    for i in range(len(vectors)):  
        ui = vectors[i]  
        for j in range(i):  
            muj = dot_product(vectors[i], u[j]) / vector_norm(u[j])**2  
            ui = [ui[k] - muj * u[j][k] for k in range(len(ui))]  
        u.append(ui)  
    return u
```

```
flag = round(gram_smith(vectors)[3][1],5)
```

```
print(flag)
```

```
gramschmidt:0.91611
```

## Inferious prime

### RSA

```
from Crypto.Util.number import getPrime, inverse, bytes_to_long, long_to_bytes, GCD
```

```
e = 3
```

```
n = 742449129124467073921545687640895127535705902454369756401331
```

```
ct = 39207274348578481322317340648475596807303160111338236677373
```

```
p = 752708788837165590355094155871
```

```
q = 986369682585281993933185289261
```

```
phi = (p-1)*(q-1)
```

```
d = inverse(e,phi)
```

```
a = pow(ct,d,n)

m = long_to_bytes(a)

print(m)

crypto{N33d_b1g_pR1m35}
```

## Square Eyes:

N =

53586080804400955002917713570816801620145134314731356537101445902774349173942288544  
30847057207314097137755279937196825836691648738068420432884398280717899706947590808  
42162253955259590552283047728782812946845160334801782088068154453021936721710269050  
98580505469209673877732179615338402489761559449345306813834120367374951409454600025  
36319029916171978475845196941521227654069821335265949286852323819347421521958613802  
21224370858128736975959176861651044370378539093990198336298572944512738570839396588  
59009681321779119189594138046480337760277924066313383495232931686239958195059058800  
63712213341282154091976032369425976747567282122321340565627163991550801088811059527  
68189193728827484667349378091100068224404684701674782399200373192433062767622841264  
05542603534976901811729962055480390249043233960056643224679581816746091618064739416  
91576472456035556927356308621487154287912427647994698969247534705398570807671700527  
83918273180304835318388177089674231640910337743789750979216202573226794240332797892  
86827630940025392593222389553071416964811656901358164319234193180078525471508329452  
63259802472192183641188778648920681859055874109771527379363107347122769566631921824  
87672474651103240004173381041237906849437490609652395748868434296753449

e = 65537

c =

22250288597418242950094838984056341529153472689135457390732951255643963281092192790  
52204867278074366680359293024427542259527866024922504480203412177336464729822862223  
38860566076161977786095675944552232391481278782019346283900959677167026636830252067  
04875972025167181105864756972449554794096688502562980707917121837164452805356223239  
66742837453101322424923672741846678451745144668341325899713880670769805631885133336  
61165819462428837210575342101036356974189393390097403614434491507672459254969638032  
77689741767457748777575553996491503573198849998372643500500785087600023229245855457  
74377394273134536714929566681882196006333259309817481624559650932226481731347775715  
27681591366164711307355510889316052064146089646772869610726671696699221157985834325  
6636614000348314424312091234787780782558468305222639096411981878490333020048870521  
27655691634955718514593555203989282142062850808839548818886685092624554908892838625  
6045359866291952224935145694435885396500780651530829377030371611921181207362217397  
80530396211210019078376306190994588971787839774071134011431159793472467060199273752  
66689328714362261353938728816645112227895652560591380026514038754849207113165225362  
60604255269532161594824301047729082877262812899724246757871448545439896

```
## from factor.db we get N's square root
```

```
p =
```

```
23148667521998097720857168827790771337662483716348435477360567409355026169165934446  
94980966459552377085389720310375910698398511326404905741690819116672000850327595162  
57389756660190291723776531706024403735795932925765306677739514076472227577564378672  
16095193174201323278896027294517792607881861855264600525772460745259440301156930943  
25524091568571855233419223026478035579917903781602633070542248400008654236208400695  
81585503463959418623839259420337300300046063603083797762554362064405294417118592468  
11586652746028418496020145441513037535475380962562108920699929022900677901988508936  
509354385660735694568216631382653107
```

```
phi = (p)*(p-1)
```

```
from Crypto.Util.number import getPrime, inverse, bytes_to_long, long_to_bytes, GCD
```

```
d = inverse(e,phi)
```

```
m = pow(c,d,N)
```

```
print(m)
```

```
a = long_to_bytes(m)
```

```
print(a)
```

```
crypto{squar3_r00t_i5_f4st3r_th4n_f4ct0r1ng!}
```