rsa

入门关：

```
x=chr(random.randint(0,0xff))+chr(random.randint(0,0xff))+chr(random.randint
(0,0x1f))
hashlib.sha256(x).hexdigest()[0:8]=='e662a10f'
@ x.encode('hex')=?
```

三个rand的范围分别为0-0xff,0-0xff,0-0x1f,暴力碰撞即可，三次循环。

```
def proof_of_work(io):
    io.read_until(b"hashlib.sha256(x).hexdigest()[0:8]=='")
    first8 = io.read(8)

    for i in range(0xff+1):
        for j in range(0xff+1):
            for k in range(0x1f+1):
                x = chr(i) + chr(j) + chr(k)
                if hashlib.sha256(x).hexdigest()[:8] == first8:
                    io.writeline(x.encode('hex'))
                    return True

    return False
```

第一关：

```
=next-rsa=
# n=0xc4606b153b9d06d934c9ff86a3be5610266387d82d11f3b4e354b1d95fc7e577
# e=0x10001
# c=0x3285835a3f730cee5c1a61f77d57e84c4c9a138bf7904485c3a41ab6f746363d
```

n比较小，所以直接分解即可，使用RSATools等工具。

第二关：

```
=next-rsa=
# n=0x92411fa0c93c1b27f89e436d8c4698bcf554938396803a5b62bd10c9bfcbf85a483bd8
7bb2d6a8dc00c32d8a7caf30d8899d90cb8f5838cae95f7ff5358847db1244006c140edfcc36
adbdcaa16cd27432b4d50d2348b5c15c209364d7914ef50425e4c3da07612cc34e9b93b98d39
4b43f3eb0a5a806c70f06697b6189606eb9707104a7b6ff059011bac957e2aae9ec406a4ff8f
8062400d2312a207a9e018f4b4e961c943dfc410a26828d2e88b24e4100162228a5bbf0824cf
2f1c8e7b915efa385efeb505a9746e5d19967766618007ddf0d99525e9a41997217484d64c6a
879d762098b9807bee46a219be76941b9ff31465463981e230eecec69691d1
# e=0x6f6b385dd0f06043c20a7d8e5920802265e1baab9d692e7c20b69391cc5635dbcaae59
726ec5882f168b3a292bd52c976533d3ad498b7f561c3dc01a76597e47cfe60614f247551b3d
be200e2196eaa001a1d183886eeacddfe82d80b38aea24de1a337177683ed802942827ce4d28
e20efef92f38f1b1a18c66f9b45f5148cceabfd736de8ac4a49e63a8d35a83b664f9f3b00f82
2b6f11ff13257ee6e0c00ca5c98e661ea594a9e66f2bd56b33d9a13f5c997e67a37fcf9a0c7f
04d119fe1ba261127357e64a4b069aefed3049c1c1fe4f964fd078b88bedd064abea385cfebd
65e563f93c12d34eb6426e8aa321033cfd8fe8855b9e74d07fe4f9d70de46f
# c=0x11cd8f66ae8ad8dc3af3685b0589cffb31327068df37b7aaa19766557beaeca5c75a5d
aa03b92eaf5e77b0c6ad685aa2245e22e813369590e574d54fe46e47676fc64a3aa5631805c6
b4223f5a34f9b4cb5f1b87565467ddfde718c606c7df822718f467dd2966da007ad964959f52
b47fe02ee551be68a246233a78e9387edcbdcb84c27f73398bf6801c1bb03cf96ddb297907e6
5811d9ecad486c24b99e4e1a67debe4c1503cf8fb6f78a2aa4c76790fa33952bca4dd06b673f
5a60141565fd00059c1b4a3def7faf2bfcf95aa70ff1e29c05db344ae27244af0036c52a1945
46521faa99092969e039e2186028cecb1bd81e6617dfafd98ce73aeddc7c31
```

e与n大小相似，可以使用Wiener Attack低解密指数攻击。

https://github.com/orisano/owiener

第三关

```
=next-rsa=
# n=0x79982a272b9f50b2c2bc8b862ccc617bb39720a6dc1a22dc909bbfd1243cc0a03dd406
ec0b1a78fa75ce5234e8c57e0aab49205090636 4353b06ccd45f90b7818b04be4734eeb8e859
ef92a306be105d32108a3165f96664ac1e00bba770f04627da05c3d7513f5882b280774 6090c
ebbf74cd50c0128559a2cc9fa7d88f7b2d
# e=0x3
# c=0x381db081852c92d268b49a1b9486d724e4ecf49fc97dc5f20d1fad902b5cdfb49c8cc1
e968e36f65ae9af7e8186f15ccdca798786669a3d2c9fe8767a7ae938a4f9115ae8fed4928d9
5ad550fddd3a9c1497785c9e2279edf43f04601980aa28b3b52afb55e2b34e5b175af25d5b3b
d71db88b3b31e48a177a469116d957592c
# b=0xfedcba9876543210000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
# m=b+x (x:64bit)
```

e=3,典型的低加密型RSA，可以使用低加密指数攻击

```
assert pow(m, e, n) == c
```

第四关

```
=next-rsa=
# n=0x78e2e04bdc50ea0b297fe9228f825543f2ee0ed4c0ad94b6198b672c3b005408fd8330
c36f55d36fb129d308c23e5cb8f4d61aa7b058c23607cef83d63c4ed0f066fc0b3c0062a2ac6
8c75ca8035b3bd7a320bdf29cfcf6cc30377743d2a8cc29f7c588b8043412366ab69ec824309
cb1ef3851d4fb14a1f0a58e4a1193f5518fa1d0c159621e1f832b474182593db2352ef05101b
f367865ad26efe14fce977e9e48d3310a18b67991958d1a01bd0f3276a669866f4deaef2a68b
faefd35fe2ba5023a22c32ae8b2979c26923ee3f855363f18d8d58bb1bc3b7f585c9d9f6618c
727f0f7b9e6f32af2864a77402803011874ed2c65545ced72b183f5c55d4d1
# e=0x10001
# nextprime(p)*nextprime(q)=0x78e2e04bdc50ea0b297fe9228f825543f2ee0ed4c0ad94
b6198b672c3b005408fd8330c36f55d36fb129d308c23e5cb8f4d61aa7b058c23607cef83d63
c4ed0f066fc0b3c0062a2ac68c75ca8035b3bd7a320bdf29cfcf6cc30377743d2a8cc29f7c58
8b8043412366ab69ec824309cb1ef3851d4fb14a1f0a58e4a1193f5a58ee70a59ac06b64dbe0
4b876ff69436b78cf03371f2062707897bf4e580870e42b5e62709b69f6d4939ac5641ea0f29
de44aaee8f2fcd0f66aaa720b584f7c801e52ce7cd41db45ceb99ebd7b51bef8d0cd2deb5c50
b59f168276c9c98d46a1c37bd3d6ef81f2c6e89028680a172e00d92dd8b392135112dd16efab
57d00b26b9
# c=0x2e1f9e68e6b125b87ace75943d721911edb498351f500d62a5a0bd926bcd97283e6294
e15c05c375b4086adfbc55262cba392a82be4f2927485d8070d7364ccaa9ff65dff9a9408b6f
ef696ce4951dc41ae6b4536e0e6afedf7d73952aa13be83385ea61780db94f3e94634730232c
9bc53cf980fc78e0cdace398d77dd38bd7852e9cbb47ca5824c35072caa86be64006da4192c2
a405576bd649ebeaaf7a88a61b990eae9be59c95946f1c34b49fa292e3782f8ee35219529190
ea774f102b5ced5291da00ef6acb80c513969ce39660db2ed1f52c4523b3c764a9acdd1fa14c
53c43378e81a8f7d2836be9973e2c75b910ee5de6c963dafa7c3688405af7e
@ m=0x80409f2902b6
```

```
n=0x78e2e04bdc50ea0b297fe9228f825543f2ee0ed4c0ad94b6198b672c3b005408fd8330c3
6f55d36fb129d308c23e5cb8f4d61aa7b058c23607cef83d63c4ed0f066fc0b3c0062a2ac68c
75ca8035b3bd7a320bdf29cfcf6cc30377743d2a8cc29f7c588b8043412366ab69ec824309cb
1ef3851d4fb14a1f0a58e4a1193f5518fa1d0c159621e1f832b474182593db2352ef05101bf3
67865ad26efe14fce977e9e48d3310a18b67991958d1a01bd0f3276a669866f4deaef2a68bfa
efd35fe2ba5023a22c32ae8b2979c26923ee3f855363f18d8d58bb1bc3b7f585c9d9f6618c72
7f0f7b9e6f32af2864a77402803011874ed2c65545ced72b183f5c55d4d1

next_n = 0x78e2e04bdc50ea0b297fe9228f825543f2ee0ed4c0ad94b6198b672c3b005408f
d8330c36f55d36fb129d308c23e5cb8f4d61aa7b058c23607cef83d63c4ed0f066fc0b3c0062
```

```
a2ac68c75ca8035b3bd7a320bdf29cfcf6cc30377743d2a8cc29f7c588b8043412366ab69ec8
24309cb1ef3851d4fb14a1f0a58e4a1193f5a58ee70a59ac06b64dbe04b876ff69436b78cf03
371f2062707897bf4e580870e42b5e62709b69f6d4939ac5641ea0f29de44aaee8f2fcd0f66a
aa720b584f7c801e52ce7cd41db45ceb99ebd7b51bef8d0cd2deb5c50b59f168276c9c98d46a
1c37bd3d6ef81f2c6e89028680a172e00d92dd8b392135112dd16efab57d00b26b9

def isqrt(n):
    x = n
    y = (x + 1) // 2
    while y < x:
        x = y
        y = (x + n // x) // 2
    return x

delta_n = next_n - n

for a in range(1, 8192*2):
    if a % 100 == 0:
        print('searching... a = %d' % a)
    for b in range(1, 8192*2):
        delta_n_minus_ab = delta_n - a * b
        squared = delta_n_minus_ab * delta_n_minus_ab - 4 * b * a * n
        if not gmpy.is_square(squared):
            continue
        s_root = int(gmpy.sqrt(squared))
        if (delta_n_minus_ab + s_root) % (2 * a) == 0:
            q = (delta_n_minus_ab + s_root) // (2 * a)
            if gmpy.is_prime(q):
                print('Found! ', a, b, q, n % q == 0)
        if (delta_n_minus_ab - s_root) % (2 * a) == 0:
            q = (delta_n_minus_ab - s_root) // (2 * a)
            if gmpy.is_prime(q):
                print('Found! ', a, b, q, n % q == 0)
```

第五关

```
=next-rsa=
# n=0x163323fcb69e36f877a69deaf1c94952e2a6ba8bd2df00271996274d1b046aa23d7fac
e7577e397ee8af6f68e2c3cd9512062449b8ff6c4aa4d36ce623405acf7e4ca1836550c97d61
bb70493dd953c56fe1557cf55a49635b68cea434270d87decc318bf188121565782274bd4aac
282d47b0c453b4cdeca6a618339f9313cd44df4e59
# e=0x10001
# c=0xc2966e700fdccb69315433122820b0b692dd25efafb814caad0c22cfd3aa393a1a8cf0
1fe9c1b5a6e3d7c1668a29a44c17a2476cef79c4b2b8b5e513dec04f142aad23ce91a129ba9f
```

```
73dfab132d1ca86e931bbd06377f890c6ea7b14e4c58ac041eb8b73fabf63efbfd85ba999c4e
f38457a4743c933b6f628027ca640b5c4dfbaac46
```

没有什么窍门，暴力分解n，p为十亿数量级，可以爆破。

第六关

```
 =next-rsa=
# n=0x7003581fa1b15b80dbe8da5dec35972e7fa42cd1b7ae50a8fc20719ee641d608098012
5d18039e95e435d2a60a4d5b0aaa42d5c13b0265da4930a874ddadcd9ab0b02efcb4463a3336
1a84df0c02dfbd05c0fdc01e52821c683bd265e556412a3f55e49517778079cb1c1c1c22ef8a
6e0bccd5e78888ff46167a471f6bff25664a34311c5cb8d6c1b1e7ac2ab0e6676d594734e8f7
013b33806868c151316d0cf762a50066c596244fd70b4cb021369aae432e174da502a806e7a8
ab13dad1f1b83ac73c0e9e39648630923cbd5726225f17cc0d15afadb7d2c2952b6e092ffc53
dcff2914bfddedd043bbdf9c6f6b6b5a6269c5bd423294b9deac4f268eaadb
# e=0x3
# c=0xb2ab05c888ab53d16f8f7cd39706a15e51618866d03e603d67a270fa83b16072a35b52
06da11423e4cd9975b4c03c9ee0d78a300df1b25f7b69708b19da1a5a570c824b2272b163de2
5b6c2f358337e44ba73741af708ad0b8d1d7fa41e24344ded8c6139644d84dc810b38450454a
f3e375f68298029b7ce7859f189cdae6cfaf166e58a22fe5a751414440bc6bce5ba580fd210c
4d37b97d8f5052a69d31b275c53b7d61c87d8fc06dc713e1c1ce05d7d0aec710eba2c1de6151
c84d7bc3131424344b90e3f8947322ef1a57dd3a459424dd31f65ff96f5b8130dfd33111c59f
3fc3a754e6f98a836b4fc6d21aa74e676f556aaa5a703eabe097140ec9d98
@ m=0xcf54ad6301f83d4c7a151d7706739935471171f3c67d13850ae75118f13f5531eef5ef
2ebf58277c22b5d89476d713e3a697d7cd71f2ac23671bb78053fdeeff1b372d7f31946568b5
bbb04140ad25d6212dd9c9e9e7
```

同第三关，低加密指数攻击，(e=3基本都是低加密指数攻击)

```
import gmpy

n = 0x7003581fa1b15b80dbe8da5dec35972e7fa42cd1b7ae50a8fc20719ee641d608098012
5d18039e95e435d2a60a4d5b0aaa42d5c13b0265da4930a874ddadcd9ab0b02efcb4463a3336
1a84df0c02dfbd05c0fdc01e52821c683bd265e556412a3f55e49517778079cb1c1c1c22ef8a
6e0bccd5e78888ff46167a471f6bff25664a34311c5cb8d6c1b1e7ac2ab0e6676d594734e8f7
013b33806868c151316d0cf762a50066c596244fd70b4cb021369aae432e174da502a806e7a8
ab13dad1f1b83ac73c0e9e39648630923cbd5726225f17cc0d15afadb7d2c2952b6e092ffc53
dcff2914bfddedd043bbdf9c6f6b6b5a6269c5bd423294b9deac4f268eaadb

c = 0xb2ab05c888ab53d16f8f7cd39706a15e51618866d03e603d67a270fa83b16072a35b52
06da11423e4cd9975b4c03c9ee0d78a300df1b25f7b69708b19da1a5a570c824b2272b163de2
5b6c2f358337e44ba73741af708ad0b8d1d7fa41e24344ded8c6139644d84dc810b38450454a
f3e375f68298029b7ce7859f189cdae6cfaf166e58a22fe5a751414440bc6bce5ba580fd210c
```

```
4d37b97d8f5052a69d31b275c53b7d61c87d8fc06dc713e1c1ce05d7d0aec710eba2c1de6151
c84d7bc3131424344b90e3f8947322ef1a57dd3a459424dd31f65ff96f5b8130dfd33111c59f
3fc3a754e6f98a836b4fc6d21aa74e676f556aaa5a703eabe097140ec9d98

for k in range(1000000):
    s = gmpy.mpz(c + n * k)
    m = int(s.root(3)[0])
    if m ** 3 == s:
        print('Found!', m)
        break
```

第七关

```
=next-rsa=
# n1=0xb4e9991d2fac12b098b01118d960eb5470261368e7b1ff2da2c66b430283 5aa845dd5
0a4f749fea749c6d439156df6faf8d14ce2a57da3bac542f1843bfc80dfd632e7a2ef96496a6
60d8c5994aea9e1b665097503558bc2756ab06d362abe3777d8c1f388c8cd1d193955b700533
82d330125bdc2cdc836453f1a26cec1021cbb787977336b2300f38c6ba881a93d2a2735f8f0d
32ea2d0e9527eb15294dd0867c8030d1f646bd121c01706c247cd1bf4aa209d383ffb748b73e
c1688dc71812675834b4b12d27a63b5b8fcc47394d16897ff96af49f39d8d5b247553fbf8fac
7be08aab43d9ce5659cd5cfaf7d73edbcfe854d997ae4b28d879adf86641707
# e1=0x10001
# c1=0x724295620055a3db8b87d83cae500eff2d6176c7120ec249a464628ad115daa110a75
61cd75eea1ea034e9b5bf9faf23c99b9a8b712d2616ac084a0cd7cf3c10f3d49103e859153a7
3525efa4735b86a894618fbe7af50cd260da131992708e6274368d28a7c3dc7574baae6e3790
82eb4716e7784a8844f20023409381 8b0dafdaf12ff4babda83831657e165d395e27af3a709c
126927a38fd8c07ba36c2290a1d6bdd95b5139adb1e272ab7e053812b5fe77e9b49e5faae240
cbeec0a34798762de3b99df98cb026d27421272b8021b04261f768b17242bb4dbcce016b6c43
4d09a45dedb45e59e88a7cede0808ef5a59fdad2f58934bfba36262f1f4608b
# n2=0xc31344c753e25135d5eed8febaa57dd7020b503a5569bdd4ae6747b5c36436dc1c4d7
ead77bfc1034748bcc630636bae1c8f4ca5dee8246b3d6f3e8b14e16487733b14ec8e587e07a
7a6de45859d32d241eaf7746c45ff404f1a767ab77e8493ae8141fee0bcf4e9b7c455415b694
5fa60de928b01dfa90bbf0d09194f93db7a1663121d281c908f0e38237f63c2b856f99c6029d
993f9afb5fbbb762044d97943ff34023486c4cf1db9ffdc439d9f5ff331b606374c7133d61e4
614fac3ea7faaf54563338b736282658e7925b22457709183 1351a28679a8d6f8e7ba16685b2
769bb49b79f8054b29c809d68aca0f2c5e3f1fd0e3ef6c21f756e3c44a40439
# e2=0x10001
# c2=0x3d81fac2c5f7b381e3dc2df5736209091aa12ab7a85909c5532dd8bac6274d1f4b590
15c823f41ac9a68c270d58616a7ef112226fe9f339bc1a074da28e383d301fe25a43dded5b63
debd1922bebe15278d0dedccd70abd1cd09fbd5ea53968c140c1e0757b7d2bac81cdbea2565b
65bf534bcf9ce3f4a7707a91c5fd07be98cdbba48efbcaa77fb95fa7d7b86111ea0234bee257
a50688dae8e806e803b4e8ae1bee20ad1c8cb8f59125e9a904d22420dd048a4f15f62ae0cd69
5879cce2f80ae9f79ba23a6c16fd23148242598f03bf8ec020aea65af67dee54cc2fc4e753f0
852feaf868445e6dbe705d7f1f1a5cba1b24dabd9702f18c4b4eba67a482b67
```

```
@ m1=0xc7c8f3138769
ok!
@ m2=0xab0f2908c6aa
ok!
```

给了两组n1,n2，推测这两个数有相同的p或q，gcd能在一秒内计算出两个数的最大公约数

```
n1 = 0xb4e9991d2fac12b098b01118d960eb5470261368e7b1ff2da2c66b4302835aa845dd5
0a4f749fea749c6d439156df6faf8d14ce2a57da3bac542f1843bfc80dfd632e7a2ef96496a6
60d8c5994aea9e1b665097503558bc2756ab06d362abe3777d8c1f388c8cd1d193955b700533
82d330125bdc2cdc836453f1a26cec1021cbb787977336b2300f38c6ba881a93d2a2735f8f0d
32ea2d0e9527eb15294dd0867c8030d1f646bd121c01706c247cd1bf4aa209d383ffb748b73e
c1688dc71812675834b4b12d27a63b5b8fcc47394d16897ff96af49f39d8d5b247553fbf8fac
7be08aab43d9ce5659cd5cfaf7d73edbcfe854d997ae4b28d879adf86641707
n2 = 0xc31344c753e25135d5eed8febaa57dd7020b503a5569bdd4ae6747b5c36436dc1c4d7
ead77bfc1034748bcc630636bae1c8f4ca5dee8246b3d6f3e8b14e16487733b14ec8e587e07a
7a6de45859d32d241eaf7746c45ff404f1a767ab77e8493ae8141fee0bcf4e9b7c455415b694
5fa60de928b01dfa90bbf0d09194f93db7a1663121d281c908f0e38237f63c2b856f99c6029d
993f9afb5fbbb762044d97943ff34023486c4cf1db9ffdc439d9f5ff331b606374c7133d61e4
614fac3ea7faaf54563338b736282658e7925b224577091831351a28679a8d6f8e7ba16685b2
769bb49b79f8054b29c809d68aca0f2c5e3f1fd0e3ef6c21f756e3c44a40439

def gcd(a, b):                                        # (1)
    """
    Returns the greatest commond divisor of a and b.
    Input:
        a -- an integer
        b -- an integer
    Output:
        an integer, the gcd of a and b
    Examples:
    >>> gcd(97,100)
    1
    >>> gcd(97 * 10**15, 19**20 * 97**2)              # (2)
    97L
    """
    if a < 0:  a = -a
    if b < 0:  b = -b
    if a == 0: return b
    if b == 0: return a
    while b != 0:
        (a, b) = (b, a%b)
    return a


print(gcd(n1, n2))
```

第八关

```
=next-rsa=
# c1=pow(m,e1,n),c2=pow(m,e2,n)
# n=0xace2aa1121d22a2153389fba0b5f3e24d8721f5e535ebf5486a74191790c4e3cdd0316
b72388e7de8be78483e1f41ca5c930df434379db76ef02f0f8cd426348b62c0155cdf1d51907
68f65ce23c60a4f2b16368188954342d282264e447353c62c10959fee475de08ec9873b84b58
17fecb74899bedde29ef1220c78767f4de11ef1756404494ae1ce4af184cbc1c7c6de8e9cd16
f814bca728e05bc56b090112f94fff686bf8122a3b199eb41080860fa0689ed7dbc8904184fb
516b2bbf6b87a0a072a07b9a26b3cda1a13192c03e24dec8734378d10f992098fe88b526ce70
876e2c7b7bd9e474307dc6864b4a8e36e28ce6d1b43e3ab5513baa6fa559ff
# e1=0xac8b
# c1=0x75f092a409953943ca7fa2787bc8f2cb2e83568f13558d7e8836fa1c0e9e59e057b81
9defad9d7226ac92e231218499401c795762385a794cec757cd4482764fb4d0936d21477df89
82c86464444776c4b370c7dfecac849dfa3ada96cbadaa94068028787a8cd22b4c1e2abb4582
8b59cbfdeeec2c52deef6dd5f5d79791b39442e859f3e12914ad5e6aa5515a73e4ed9da01052
acc44b9c33fe5879a09bc70a593749a420c86962ffb0e6b5ca6b0a54fd66a86ccb238093bbfd
1f082d1ab53aaab8b1dbd58fdc7ae0613c3290a6a040c5915cabf6d6fcc67908abbb3317cfec
841f4e8c3311a3b4e236b47afb40aabc973a3ef10e2c42b5941da66bc2aff59
# e2=0x1091
# c2=0x7f6df58661f6b5b60bf999b587fbfc5279f5320e08f8db898833f155ab0cc27f52673
d64f1ccfdad867c13f5d0cde70dc05e50ee0c0600d5588d02f605e8cad8c6f714db705f2137d
ea35ed701ccb04990e990047fa7e909482b58cd5a3ce137720d1bd3a966020d8abebc608d491
4392b92bc637558fc17864638e05760e7dd011eaaf3fa68a983037bbee20d61f4f001e3cf466
dd66b3a7ea4a0b92ad128aa276b295052a08f858806b863e414f7e09317c7a3ff52189e5e727
0bdad07b799568799d91e70c390b293096fdf0be9a7927252c5f312f56ed468ea6a79bd64623
caf8a5a5eb9d89c285d9a8cae9e5d50d442074c14cdcb9f04eebf0952db7d08
@ m=0xe00fae5d435a
ok!
```

共模攻击，下式通过不断地让互质的e1,e2相互消减，直到e1=1，m^{e1} %n=c1=m
其中

```python
def find_m(n, e1, c1, e2, c2):
    while True:
        if e1 > e2:
            e1, e2 = e2, e1
            c1, c2 = c2, c1
        if e1 == 1:
            return c1
        e2 = e2 - e1
        c2 = (c2 * element_number_theory.inversemod(c1, n)) % n
        #式中c2=m^(e2-e1)%n
m8 = find_m(n8, e8_1, c8_1, e8_2, c8_2)
io.writeline(hex(m8).replace('L', ''))
```

第九关

```
=next-rsa=
# c1=pow(m,e,n1),c2=pow(m,e,n2),c3=pow(m,e,n3)
# e=0x3
# n1=0x43d819a4caf16806e1c540fd7c0e51a96a6dfdbe68735a5fd99a468825e5ee55c4087
106f7d1f91e10d50df1f2082f0f32bb82f398134b0b8758353bdabc5ba2817f4e6e0786e1766
86b2e75a7c47d073f346d6adb2684a9d28b658dddc75b3c5d10a22a3e85c6c12549d0ce7577e
79a068405d3904f3f6b9cc408c4cd8595bf67fe672474e0b94dc99072caaa4f866fc6c3feddc
74f10d6a0fb31864f52adef71649684f1a72c910ec5ca7909cc10aef85d43a57ec91f096a2d4
794299e967fcd5add6e9cfb5baf7751387e24b93dbc1f37315ce573dc063ecddd4ae6fb91273
07cfc80a037e7ff5c40a5f7590c8b2f5bd06dd392fbc51e5d059cffbcb85555
# c1=0x938eea684085e1f2a3202c47e3ab7f2b6711c8eaa8f5a73d96f377ad03d6ed391f65f
69e615d85d321518ed7e58120971e53d51ff0c45d31fad27d9fcf681c5d4d798deabc294681b
2be2d3ac6c89051dcbae58fd20112086675ca4d9eae4be192059b3981449505c401cb947ef34
d553a1d5a36271ad26283fff04e4fc44fffe10d4b91bc28124238900 8aae8a3aff73f31f748
6ea6b39ff78edc0fc3acadda0bae719144ca9df18019179b857630b72dda13d240987a8d4d8d
e48b10dac1d6f275767c19922896e0b4b7643005914f13cb400ceb18327f41cf73fbb306ad63
a7bcd56971c39106f99cc5a206641afb6c2221bf32a89cdb7c496bff7d47a1
# n2=0x60d175fdb0a96eca160fb0cbf8bad1a14dd680d353a7b3bc77e620437da70fd9153f7
609efde652b825c4ae7f25decf14a3c8240ea8c5892003f1430cc88b0ded9dae12ebffc6b236
32ac530ac4ae23fbffb7cfe431ff3d802f5a54ab76257a86aeec1cf47d482fec970fc27c5b37
6fbf2cf993270bba9b78174395de3346d4e221d1eafdb8eecc8edb953d1ccaa5fc250aed83b3
a458f9e9d947c4b01a6e72ce4fee37e77faaf5597d780ad5f0a7623edb08ce76264f72c3ff17
afc932f5812b10692bcc941a18b6f3904ca31d038baf3fc1968d1cc0588a656d0c53cd5c89ce
dba8a5230956af2170554d27f524c2027adce84fd4d0e018dc88ca4d5d26867
# c2=0x4e4f30e8230a68c4cb590985a1c96c7606a2825d534c2476ab60bb896c03de0cd13aa
a998c2f2ce4071d0f359397cb9f60056e21803a54a929720555dd7364b90102650f03e579712
48ef86f8b6e487332e4d240a433a12395dc90c6ca1ebad98a85b5184d4e8b139df6b43189935
a05fb21303b6cc515d6c0b1464d3628e724cd4b0ab4923fb7ea378331c1f78b6ed74663f704a
1291c0c1a03e0ea6e737ebe46f5a4ef17758f2c6596fdcc6f8516d0464fa3ae797349d093ed4
2e5d365cc61f71e7ad353fd4ddf635311dfa49e822b1fbb10faee1b0f8d006b016241ce5ed20
9f5ae91913a76bcec9e8ecfb40f4a2ce3baae625c0b896f991c1564467d321f
# n3=0x280f992dd63fcabdcb739f52c5ed1887e720cbfe73153adf5405819396b28cb54423d
196600cce76c8554cd963281fc4b153e3b257e96d091e5d99567dd1fa9ace52511ace4da407f
5269e71b1b13822316d751e788dc935d63916075530d7fb89cbec9b02c01aef19c39b4ecaa1f
7fe2faf990aa938eb89730eda30558e669da5459ed96f1463a983443187359c07fba8e970244
52087b410c9ac1e39ed1c74f380fd29ebdd28618d60c36e6973fc87c066cae05e9e270b5ac25
ea5ca0bac5948de0263d8cc89d91c4b574202e71811d0ddf1ed23c1bc35f3a042aac6a0bdf32
```

```
d37dede3536f70c257aafb4cfbe3370cd7b4187c023c35671de3888a1ed1303
# c3=0x19a7b73d90a153c48e3133767c78ae651108bbc3e90df6bd4b85542ea9c67a6a9588f
0fac3b31ba1adfe48de707d3e7d145c1c31e79a0fb1baac4de3f79de1a9a71d47541dcd8e151
15a83cc6fd1c331f7f24081fe22a30492f3952ac0717289e8173a26d3aee97706e2137310b66
1477a5b7cf587d62f4bab97779dbebe64600efac17ed9998ebcd180f48b491188acc9deb9d91
b982f999463f45c6acd0623d256dc7b6935dc2a7b8632bdc73721876697c716af58d7cecdd9e
f4cee7ee8832d2d6ece419835b06147c46646530dee83ece9373ddd58f6f8ac315249ee4d63a
8ea9b2153989af0ef335068ba94ca17a913e2e57b3256abc3d1e3a26edf893
@ m=0x1000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000004cb1a051687a
```

中国剩余定理：
有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二。问物几何？即，一个整数除以三余二，除以五余三，除以七余二，求这个整数。《孙子算经》中首次提到了同余方程组问题，以及以上具体问题的解法，因此在中文数学文献中也会将中国剩余定理称为孙子定理。

```python
def chinese_remainder(n, a):
    sum = 0
    prod = reduce(lambda a, b: a*b, n)

    for n_i, a_i in zip(n, a):
        p = prod / n_i
        sum += a_i * mul_inv(p, n_i) * p
    return sum % prod

def mul_inv(a, b):
    b0 = b
    x0, x1 = 0, 1
    if b == 1: return 1
    while a > 1:
        q = a / b
        a, b = b, a%b
        x0, x1 = x1 - q * x0, x0
    if x1 < 0: x1 += b0
    return x1

import gmpy
m9_cubic = chinese_remainder([n9_1, n9_2, n9_3], [c9_1, c9_2, c9_3])
m9 = int(gmpy.mpz(m9_cubic).root(3)[0])
```

日志

```
====next-rsa====
teamtoken:icqaf8ed5af3b2c03367db8ee4664528
ok!
Firstly, please give me the proof of your work!
x=chr(random.randint(0,0xff))+chr(random.randint(0,0xff))+chr(random.randint
(0,0x1f))
hashlib.sha256(x).hexdigest()[0:8]=='92d91297'
@ x.encode('hex')=da7418
ok!


input format:almost hex(m).replace("L","")

=next-rsa=
# n=0xc4606b153b9d06d934c9ff86a3be5610266387d82d11f3b4e354b1d95fc7e577
# e=0x10001
# c=0x3285835a3f730cee5c1a61f77d57e84c4c9a138bf7904485c3a41ab6f746363d
@ m=0xa1a7db40d4ff
ok!
=next-rsa=
# n=0x92411fa0c93c1b27f89e436d8c4698bcf554938396803a5b62bd10c9bfcbf85a483bd8
7bb2d6a8dc00c32d8a7caf30d8899d90cb8f5838cae95f7ff5358847db1244006c140edfcc36
adbdcaa16cd27432b4d50d2348b5c15c209364d7914ef50425e4c3da07612cc34e9b93b98d39
4b43f3eb0a5a806c70f06697b6189606eb9707104a7b6ff059011bac957e2aae9ec406a4ff8f
8062400d2312a207a9e018f4b4e961c943dfc410a26828d2e88b24e4100162228a5bbf0824cf
2f1c8e7b915efa385efeb505a9746e5d19967766618007ddf0d99525e9a41997217484d64c6a
879d762098b9807bee46a219be76941b9ff31465463981e230eecec69691d1
# e=0x6f6b385dd0f06043c20a7d8e5920802265e1baab9d692e7c20b69391cc5635dbcaae59
726ec5882f168b3a292bd52c976533d3ad498b7f561c3dc01a76597e47cfe60614f247551b3d
be200e2196eaa001a1d183886eeacddfe82d80b38aea24de1a337177683ed802942827ce4d28
e20efef92f38f1b1a18c66f9b45f5148cceabfd736de8ac4a49e63a8d35a83b664f9f3b00f82
2b6f11ff13257ee6e0c00ca5c98e661ea594a9e66f2bd56b33d9a13f5c997e67a37fcf9a0c7f
04d119fe1ba261127357e64a4b069aefed3049c1c1fe4f964fd078b88bedd064abea385cfebd
65e563f93c12d34eb6426e8aa321033cfd8fe8855b9e74d07fe4f9d70de46f
# c=0x11cd8f66ae8ad8dc3af3685b0589cffb31327068df37b7aaa19766557beaeca5c75a5d
aa03b92eaf5e77b0c6ad685aa2245e22e813369590e574d54fe46e47676fc64a3aa5631805c6
b4223f5a34f9b4cb5f1b87565467ddfde718c606c7df822718f467dd2966da007ad964959f52
b47fe02ee551be68a246233a78e9387edcbdcb84c27f73398bf6801c1bb03cf96ddb297907e6
5811d9ecad486c24b99e4e1a67debe4c1503cf8fb6f78a2aa4c76790fa33952bca4dd06b673f
5a60141565fd00059c1b4a3def7faf2bfcf95aa70ff1e29c05db344ae27244af0036c52a1945
46521faa99092969e039e2186028cecb1bd81e6617dfafd98ce73aeddc7c31
@ m=0x4ae3c3ddf3d9
ok!
=next-rsa=
# n=0x79982a272b9f50b2c2bc8b862ccc617bb39720a6dc1a22dc909bbfd1243cc0a03dd406
ec0b1a78fa75ce5234e8c57e0aab49205090636 4353b06ccd45f90b7818b04be4734eeb8e859
ef92a306be105d32108a3165f96664ac1e00bba770f04627da05c3d7513f5882b280774609 0c
ebbf74cd50c0128559a2cc9fa7d88f7b2d
```

```
# e=0x3
# c=0x381db081852c92d268b49a1b9486d724e4ecf49fc97dc5f20d1fad902b5cdfb49c8cc1
e968e36f65ae9af7e8186f15ccdca798786669a3d2c9fe8767a7ae938a4f9115ae8fed4928d9
5ad550fddd3a9c1497785c9e2279edf43f04601980aa28b3b52afb55e2b34e5b175af25d5b3b
d71db88b3b31e48a177a469116d957592c
# b=0xfedcba9876543210000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
# m=b+x (x:64bit)
@ x=0x33686739766d336b
ok!
=next-rsa=
# n=0x78e2e04bdc50ea0b297fe9228f825543f2ee0ed4c0ad94b6198b672c3b005408fd8330
c36f55d36fb129d308c23e5cb8f4d61aa7b058c23607cef83d63c4ed0f066fc0b3c0062a2ac6
8c75ca8035b3bd7a320bdf29cfcf6cc30377743d2a8cc29f7c588b8043412366ab69ec824309
cb1ef3851d4fb14a1f0a58e4a1193f5518fa1d0c159621e1f832b474182593db2352ef05101b
f367865ad26efe14fce977e9e48d3310a18b67991958d1a01bd0f3276a669866f4deaef2a68b
faefd35fe2ba5023a22c32ae8b2979c26923ee3f855363f18d8d58bb1bc3b7f585c9d9f6618c
727f0f7b9e6f32af2864a77402803011874ed2c65545ced72b183f5c55d4d1
# e=0x10001
# nextprime(p)*nextprime(q)=0x78e2e04bdc50ea0b297fe9228f825543f2ee0ed4c0ad94
b6198b672c3b005408fd8330c36f55d36fb129d308c23e5cb8f4d61aa7b058c23607cef83d63
c4ed0f066fc0b3c0062a2ac68c75ca8035b3bd7a320bdf29cfcf6cc30377743d2a8cc29f7c58
8b8043412366ab69ec824309cb1ef3851d4fb14a1f0a58e4a1193f5a58ee70a59ac06b64dbe0
4b876ff69436b78cf03371f2062707897bf4e580870e42b5e62709b69f6d4939ac5641ea0f29
de44aaee8f2fcd0f66aaa720b584f7c801e52ce7cd41db45ceb99ebd7b51bef8d0cd2deb5c50
b59f168276c9c98d46a1c37bd3d6ef81f2c6e89028680a172e00d92dd8b392135112dd16efab
57d00b26b9
# c=0x2e1f9e68e6b125b87ace75943d721911edb498351f500d62a5a0bd926bcd97283e6294
e15c05c375b4086adfbc55262cba392a82be4f2927485d8070d7364ccaa9ff65dff9a9408b6f
ef696ce4951dc41ae6b4536e0e6afedf7d73952aa13be83385ea61780db94f3e94634730232c
9bc53cf980fc78e0cdace398d77dd38bd7852e9cbb47ca5824c35072caa86be64006da4192c2
a405576bd649ebeaaf7a88a61b990eae9be59c95946f1c34b49fa292e3782f8ee35219529190
ea774f102b5ced5291da00ef6acb80c513969ce39660db2ed1f52c4523b3c764a9acdd1fa14c
53c43378e81a8f7d2836be9973e2c75b910ee5de6c963dafa7c3688405af7e
@ m=0x80409f2902b6
ok!
=next-rsa=
# n=0x163323fcb69e36f877a69deaf1c94952e2a6ba8bd2df00271996274d1b046aa23d7fac
e7577e397ee8af6f68e2c3cd9512062449b8ff6c4aa4d36ce623405acf7e4ca1836550c97d61
bb70493dd953c56fe1557cf55a49635b68cea434270d87decc318bf188121565782274bd4aac
282d47b0c453b4cdeca6a618339f9313cd44df4e59
# e=0x10001
# c=0xc2966e700fdccb69315433122820b0b692dd25efafb814caad0c22cfd3aa393a1a8cf0
1fe9c1b5a6e3d7c1668a29a44c17a2476cef79c4b2b8b5e513dec04f142aad23ce91a129ba9f
73dfab132d1ca86e931bbd06377f890c6ea7b14e4c58ac041eb8b73fabf63efbfd85ba999c4e
f38457a4743c933b6f628027ca640b5c4dfbaac46
@ m=0xd436441892a5
ok!
```

```
=next-rsa=
# n=0x7003581fa1b15b80dbe8da5dec35972e7fa42cd1b7ae50a8fc20719ee641d608098012
5d18039e95e435d2a60a4d5b0aaa42d5c13b0265da4930a874ddadcd9ab0b02efcb4463a3336
1a84df0c02dfbd05c0fdc01e52821c683bd265e556412a3f55e49517778079cb1c1c1c22ef8a
6e0bccd5e78888ff46167a471f6bff25664a34311c5cb8d6c1b1e7ac2ab0e6676d594734e8f7
013b33806868c151316d0cf762a50066c596244fd70b4cb021369aae432e174da502a806e7a8
ab13dad1f1b83ac73c0e9e39648630923cbd5726225f17cc0d15afadb7d2c2952b6e092ffc53
dcff2914bfddedd043bbdf9c6f6b6b5a6269c5bd423294b9deac4f268eaadb
# e=0x3
# c=0xb2ab05c888ab53d16f8f7cd39706a15e51618866d03e603d67a270fa83b16072a35b52
06da11423e4cd9975b4c03c9ee0d78a300df1b25f7b69708b19da1a5a570c824b2272b163de2
5b6c2f358337e44ba73741af708ad0b8d1d7fa41e24344ded8c6139644d84dc810b38450454a
f3e375f68298029b7ce7859f189cdae6cfaf166e58a22fe5a751414440bc6bce5ba580fd210c
4d37b97d8f5052a69d31b275c53b7d61c87d8fc06dc713e1c1ce05d7d0aec710eba2c1de6151
c84d7bc3131424344b90e3f894732 2ef1a57dd3a459424dd31f65ff96f5b8130dfd33111c59f
3fc3a754e6f98a836b4fc6d21aa74e676f556aaa5a703eabe097140ec9d98
@ m=0xcf54ad6301f83d4c7a151d7706739935471171f3c67d13850ae75118f13f5531eef5ef
2ebf58277c22b5d89476d713e3a697d7cd71f2ac23671bb78053fdeeff1b372d7f31946568b5
bbb04140ad25d6212dd9c9e9e7
ok!
=next-rsa=
# n1=0xb4e9991d2fac12b098b01118d960eb5470261368e7b1ff2da2c66b430283 5aa845dd5
0a4f749fea749c6d439156df6faf8d14ce2a57da3bac542f1843bfc80dfd632e7a2ef96496a6
60d8c5994aea9e1b665097503558bc2756ab06d362abe3777d8c1f388c8cd1d193955b700533
82d330125bdc2cdc836453f1a26cec1021cbb787977336b2300f38c6ba881a93d2a2735f8f0d
32ea2d0e9527eb15294dd0867c8030d1f646bd121c01706c247cd1bf4aa209d383ffb748b73e
c1688dc71812675834b4b12d27a63b5b8fcc47394d16897ff96af49f39d8d5b247553fbf8fac
7be08aab43d9ce5659cd5cfaf7d73edbcfe854d997ae4b28d879adf86641707
# e1=0x10001
# c1=0x724295620055a3db8b87d83cae500eff2d6176c7120ec249a464628ad115daa110a75
61cd75eea1ea034e9b5bf9faf23c99b9a8b712d2616ac084a0cd7cf3c10f3d49103e859153a7
3525efa4735b86a894618fbe7af50cd260da131992708e6274368d28a7c3dc7574baae6e3790
82eb4716e7784a8844f20023409381 8b0dafdaf12ff4babda83831657e165d395e27af3a709c
126927a38fd8c07ba36c2290a1d6bdd95b5139adb1e272ab7e053812b5fe77e9b49e5faae240
cbeec0a34798762de3b99df98cb026d27421272b8021b04261f768b17242bb4dbcce016b6c43
4d09a45dedb45e59e88a7cede0808ef5a59fdad2f58934bfba36262f1f4608b
# n2=0xc31344c753e25135d5eed8febaa57dd7020b503a5569bdd4ae6747b5c36436dc1c4d7
ead77bfc1034748bcc630636bae1c8f4ca5dee8246b3d6f3e8b14e16487733b14ec8e587e07a
7a6de45859d32d241eaf7746c45ff404f1a767ab77e8493ae8141fee0bcf4e9b7c455415b694
5fa60de928b01dfa90bbf0d09194f93db7a1663121d281c908f0e38237f63c2b856f99c6029d
993f9afb5fbbb762044d97943ff34023486c4cf1db9ffdc439d9f5ff331b606374c7133d61e4
614fac3ea7faaf54563338b736282658e7925b22457709183 1351a28679a8d6f8e7ba16685b2
769bb49b79f8054b29c809d68aca0f2c5e3f1fd0e3ef6c21f756e3c44a40439
# e2=0x10001
# c2=0x3d81fac2c5f7b381e3dc2df5736209091aa12ab7a85909c5532dd8bac6274d1f4b590
15c823f41ac9a68c270d58616a7ef112226fe9f339bc1a074da28e383d301fe25a43dded5b63
debd1922bebe15278d0dedccd70abd1cd09fbd5ea53968c140c1e0757b7d2bac81cdbea2565b
65bf534bcf9ce3f4a7707a91c5fd07be98cdbba48efbcaa77fb95fa7d7b86111ea0234bee257
```

```
a50688dae8e806e803b4e8ae1bee20ad1c8cb8f59125e9a904d22420dd048a4f15f62ae0cd69
5879cce2f80ae9f79ba23a6c16fd23148242598f03bf8ec020aea65af67dee54cc2fc4e753f0
852feaf868445e6dbe705d7f1f1a5cba1b24dabd9702f18c4b4eba67a482b67
@ m1=0xc7c8f3138769
ok!
@ m2=0xab0f2908c6aa
ok!
=next-rsa=
# c1=pow(m,e1,n),c2=pow(m,e2,n)
# n=0xace2aa1121d22a2153389fba0b5f3e24d8721f5e535ebf5486a74191790c4e3cdd0316
b72388e7de8be78483e1f41ca5c930df434379db76ef02f0f8cd426348b62c0155cdf1d51907
68f65ce23c60a4f2b16368188954342d282264e447353c62c10959fee475de08ec9873b84b58
17fecb74899bedde29ef1220c78767f4de11ef1756404494ae1ce4af184cbc1c7c6de8e9cd16
f814bca728e05bc56b090112f94fff686bf8122a3b199eb41080860fa0689ed7dbc8904184fb
516b2bbf6b87a0a072a07b9a26b3cda1a13192c03e24dec8734378d10f992098fe88b526ce70
876e2c7b7bd9e474307dc6864b4a8e36e28ce6d1b43e3ab5513baa6fa559ff
# e1=0xac8b
# c1=0x75f092a409953943ca7fa2787bc8f2cb2e83568f13558d7e8836fa1c0e9e59e057b81
9defad9d7226ac92e231218499401c795762385a794cec757cd4482764fb4d0936d21477df89
82c86464444776c4b370c7dfecac849dfa3ada96cbadaa94068028787a8cd22b4c1e2abb4582
8b59cbfdeeec2c52deef6dd5f5d79791b39442e859f3e12914ad5e6aa5515a73e4ed9da01052
acc44b9c33fe5879a09bc70a593749a420c86962ffb0e6b5ca6b0a54fd66a86ccb238093bbfd
1f082d1ab53aaab8b1dbd58fdc7ae0613c3290a6a040c5915cabf6d6fcc67908abbb3317cfec
841f4e8c3311a3b4e236b47afb40aabc973a3ef10e2c42b5941da66bc2aff59
# e2=0x1091
# c2=0x7f6df58661f6b5b60bf999b587fbfc5279f5320e08f8db898833f155ab0cc27f52673
d64f1ccfdad867c13f5d0cde70dc05e50ee0c0600d5588d02f605e8cad8c6f714db705f2137d
ea35ed701ccb04990e990047fa7e909482b58cd5a3ce137720d1bd3a966020d8abebc608d491
4392b92bc637558fc17864638e05760e7dd011eaaf3fa68a983037bbee20d61f4f001e3cf466
dd66b3a7ea4a0b92ad128aa276b295052a08f858806b863e414f7e09317c7a3ff52189e5e727
0bdad07b799568799d91e70c390b293096fdf0be9a7927252c5f312f56ed468ea6a79bd64623
caf8a5a5eb9d89c285d9a8cae9e5d50d442074c14cdcb9f04eebf0952db7d08
@ m=0xe00fae5d435a
ok!
=next-rsa=
# c1=pow(m,e,n1),c2=pow(m,e,n2),c3=pow(m,e,n3)
# e=0x3
# n1=0x43d819a4caf16806e1c540fd7c0e51a96a6dfdbe68735a5fd99a468825e5ee55c4087
106f7d1f91e10d50df1f2082f0f32bb82f398134b0b8758353bdabc5ba2817f4e6e0786e1766
86b2e75a7c47d073f346d6adb2684a9d28b658dddc75b3c5d10a22a3e85c6c12549d0ce7577e
79a068405d3904f3f6b9cc408c4cd8595bf67fe672474e0b94dc99072caaa4f866fc6c3feddc
74f10d6a0fb31864f52adef71649684f1a72c910ec5ca7909cc10aef85d43a57ec91f096a2d4
794299e967fcd5add6e9cfb5baf7751387e24b93dbc1f37315ce573dc063ecddd4ae6fb91273
07cfc80a037e7ff5c40a5f7590c8b2f5bd06dd392fbc51e5d059cffbcb85555
# c1=0x938eea684085e1f2a3202c47e3ab7f2b6711c8eaa8f5a73d96f377ad03d6ed391f65f
69e615d85d321518ed7e58120971e53d51ff0c45d31fad27d9fcf681c5d4d798deabc294681b
2be2d3ac6c89051dcbae58fd20112086675ca4d9eae4be192059b3981449505c401cb947ef34
d553a1d5a36271ad26283fff04e4fc44ffffe10d4b91bc281242389008aae8a3aff73f31f748
```

```
6ea6b39ff78edc0fc3acadda0bae719144ca9df18019179b857630b72dda13d240987a8d4d8d
e48b10dac1d6f275767c19922896e0b4b7643005914f13cb400ceb18327f41cf73fbb306ad63
a7bcd56971c39106f99cc5a206641afb6c2221bf32a89cdb7c496bff7d47a1
# n2=0x60d175fdb0a96eca160fb0cbf8bad1a14dd680d353a7b3bc77e620437da70fd9153f7
609efde652b825c4ae7f25decf14a3c8240ea8c5892003f1430cc88b0ded9dae12ebffc6b236
32ac530ac4ae23fbffb7cfe431ff3d802f5a54ab76257a86aeec1cf47d482fec970fc27c5b37
6fbf2cf993270bba9b78174395de3346d4e221d1eafdb8eecc8edb953d1ccaa5fc250aed83b3
a458f9e9d947c4b01a6e72ce4fee37e77faaf5597d780ad5f0a7623edb08ce76264f72c3ff17
afc932f5812b10692bcc941a18b6f3904ca31d038baf3fc1968d1cc0588a656d0c53cd5c89ce
dba8a5230956af2170554d27f524c2027adce84fd4d0e018dc88ca4d5d26867
# c2=0x4e4f30e8230a68c4cb590985a1c96c7606a2825d534c2476ab60bb896c03de0cd13aa
a998c2f2ce4071d0f359397cb9f60056e21803a54a929720555dd7364b90102650f03e579712
48ef86f8b6e487332e4d240a433a12395dc90c6ca1ebad98a85b5184d4e8b139df6b43189935
a05fb21303b6cc515d6c0b1464d3628e724cd4b0ab4923fb7ea378331c1f78b6ed74663f704a
1291c0c1a03e0ea6e737ebe46f5a4ef17758f2c6596fdcc6f8516d0464fa3ae797349d093ed4
2e5d365cc61f71e7ad353fd4ddf635311dfa49e822b1fbb10faee1b0f8d006b016241ce5ed20
9f5ae91913a76bcec9e8ecfb40f4a2ce3baae625c0b896f991c1564467d321f
# n3=0x280f992dd63fcabdcb739f52c5ed1887e720cbfe73153adf5405819396b28cb54423d
196600cce76c8554cd963281fc4b153e3b257e96d091e5d99567dd1fa9ace52511ace4da407f
5269e71b1b13822316d751e788dc935d63916075530d7fb89cbec9b02c01aef19c39b4ecaa1f
7fe2faf990aa938eb89730eda30558e669da5459ed96f1463a983443187359c07fba8e970244
52087b410c9ac1e39ed1c74f380fd29ebdd28618d60c36e6973fc87c066cae05e9e270b5ac25
ea5ca0bac5948de0263d8cc89d91c4b574202e71811d0ddf1ed23c1bc35f3a042aac6a0bdf32
d37dede3536f70c257aafb4cfbe3370cd7b4187c023c35671de3888a1ed1303
# c3=0x19a7b73d90a153c48e3133767c78ae651108bbc3e90df6bd4b85542ea9c67a6a9588f
0fac3b31ba1adfe48de707d3e7d145c1c31e79a0fb1baac4de3f79de1a9a71d47541dcd8e151
15a83cc6fd1c331f7f24081fe22a30492f3952ac0717289e8173a26d3aee97706e2137310b66
1477a5b7cf587d62f4bab97779dbebe64600efac17ed9998ebcd180f48b491188acc9deb9d91
b982f999463f45c6acd0623d256dc7b6935dc2a7b8632bdc73721876697c716af58d7cecdd9e
f4cee7ee8832d2d6ece419835b06147c46646530dee83ece9373ddd58f6f8ac315249ee4d63a
8ea9b2153989af0ef335068ba94ca17a913e2e57b3256abc3d1e3a26edf893
@ m=0x1000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000004cb1a051687a
ok!
flag{s1mp13_rs4_f0r_y0u_+_h4pp9_f0r_qwb}
```