【CTF WriteUp】2020祥云杯Crypto题解

原创

零食商人 于 2020-11-29 06:34:57 发布 2160 收藏 4

【线下混战结束，终于有时间整理WP了】

Crypto

Exposure

本题为已知dp部分高位的情况，使用通过化简构造f(x)的Coppersmith解决

```
from sage.all import *

n = 140376049134934822153964243403031201922239588054133319056483413311963385321279682186354948441840374124640187
894619689719746347334298621083485494086361152915457458004998419817456902929318697902819798254427945343361548635 7
943083628232391509192403070726886230007477811033754818345712744230048562768412256752418 63
e = 7621
c = 467359622048571905204764348988810015306657181556988988826034220234849938866885869291225041813418609545906 05
062759610506760516932202805880472336282598807124155930399775858058909200893186430025978370000496261549009085433 8
476121035883584397407296008085715072701098582769019049679320701235521460539303638880761 6
s = 115369684682371545834265856839253777817184001492374525375952943297793218332255394443023687998 5

def coppersmith(bits, k):
    F.<x> = PolynomialRing(Zmod(n))
    invE = inverse_mod(e, n)
    f = (s << bits) + x + (k - 1) * invE    # make monic
    x0 = f.small_roots(X=2 ** bits, beta=0.44, epsilon=1/32)
    return x0

for k in range(1, e):
    bits = 200
    x0 = coppersmith(bits,k)
    if len(x0) != 0:
        x = Integer(x0[0])
        dp = x + (s << bits)
        p = (e*dp - 1) // k+1
        if p != -1:
            q = n // p
            assert n == p * q
            phi = (p-1)*(q-1)
            d = inverse_mod(e,phi)
            print d
            print pow(c,d,n)
```

more_calc

本题怎么说呢。。网上充满了非预期解，因为这题实在太容易非预期了，主代码又肯定不能跑，试一试就出来了。所以这里我们讲一讲正规套路。本题改编自2012年新加坡数学奥林匹克。

题目要求计算

$$1^{p-2}+2^{p-2}+\ldots+\left(\frac{p-1}{2}\right)^{p-2} \pmod p$$

的值。注意到当i = 1, 2, …, (p-1)/2时，

$$C_p^{2i}=\frac{p(p-1)(p-2)\ldots(p-(2i-1))}{(2i)!}$$
$$\frac{2i}{p}C_p^{2i}=\frac{(p-1)(p-2)\ldots(p-(2i-1))}{(2i-1)!}$$
$$=\frac{(-1)(-2)\ldots(-(2i-1))}{(2i-1)!}=-1 \pmod p$$

所以

$$\sum_{i=1}^{\frac{p-1}{2}} i^{p-2}=-\sum_{i=1}^{\frac{p-1}{2}} i^{p-2}\cdot\frac{2i}{p}C_p^{2i}=-\frac{2}{p}\sum_{i=1}^{\frac{p-1}{2}} i^{p-1}\cdot C_p^{2i}=-\frac{2}{p}\sum_{i=1}^{\frac{p-1}{2}} C_p^{2i} \pmod p$$

其中最后一步根据费马小定理，对于素数p与小于p的数i，有i ** (p-1)=1 (mod p)。注意到

$$\sum_{i=1}^{\frac{p-1}{2}} C_p^{2i}$$

从意义上代表从p个元素中取出偶数个元素的方法。由于p是奇数，所以从p个元素中取出偶数个元素的方法数 = 从p个元素中取出奇数个元素的方法数（剩下），且二者之和为从p中取出元素的方法数2 ** p。由于没有计算取出0个元素的情况，所以此处有

$$\sum_{i=1}^{\frac{p-1}{2}} C_p^{2i}=2^{p-1}-1$$

即

$$\sum_{i=1}^{\frac{p-1}{2}} i^{p-2}=-\frac{2}{p}\cdot(2^{p-1}-1) \pmod p$$

以此方法求出q，进而求解本题。完整代码如下：

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import gmpy2
from libnum import n2s

e = 65537
p = 27405107041753266489145388621858169511872996622765267064868542117269875531364939896671662734188734825462948115530667205007939029215517180761866791579330410449202307248373229224662232822180397215721163369151115019770596528704719472424551024516928606584975793350814943997731939996459959720826025110179216477709373849945411483731524831284895024319654509286305913312306154387754998813276562173335189450448233216133842189148761197948559529960144453513191371225490203116875516512421878350474083444237936331148910873221605156695349827919853779462052180077391722800240297035808703350489720502188129515404665633586530362179306 9
c = 35055918683748883282174784323651813560520737603185800227424500428762264933021511381871995418539707283801414497303232960090541986190867832897131815320508500774326925395739528242032566313216102210036548100374594081897428098804503420454038574457280610255242042832626554192534670284369336699175346822030007088865173250252079700270724860427575514471342164997149244044205247072315311115645755855836214700200464613652201134426101746190195358346246762242881016710707928119020973125199597600335220176686188732073999025860155060600538887296782517962617671450347555788381054344555539001456268680189452831160062315698482986474322296387716709989292671747978922668181058489406663507675599642320338049377613048817085979874567772781052867215035033348050642450667612710852648837001109914769887507004392552421783737965416800917979813137835262317794775319294801257483177741372991005066875900770459644762548438474388076655842822437141772648037236281057239552272508379892613346840960049192531743799845858272389712078645821963027561694961956409973354276629777068204456160534409039477757097372521171307620184694243888389707840806777932547158990704118642378158004690358831695861544319681913385236756504946707671037639508589887549565323717819837942112908652 3
s = p - (pow(2, p, p*p)-2)//p
q = gmpy2.next_prime(s)
n = p*q
e = 0x10001
phi = (p-1)*(q-1)
d = gmpy2.invert(e, phi)
print n2s(pow(c, d, n))
```

RSAssss

n是四个数的乘积，先yafu分解一下试试，得到两个值

```
a = 8961506852753883631560212415400830028663693459961733486750905307662271536580937174003731655887179643390684446407099586929365408257788757819718240804517503533928508572800283822031406847467097522877846424008808433180742072012136448676501116966974755339366165091211422822730857994016426987710197372845225287938 3
b = 8961506852753883631560212415400830028663693459961733486750905307662271536580937174003731655887179643390684446407099586929365408257788757819718240804517278179870317365057473764491451559152225675884808995557871345871523453666441521652683096783186230151863658670221218908795913650933410277285565766409157063007 9
```

再往下分解不下去了，所以这两个应该分别是p * next_prime(q)和q * next_prime(p)。注意到next_prime(p)有一个性质，就是当p较大时，并不会完整验算，所以next_prime(p) - p实际上很小。设

```
next_prime(q) - q = x
next_prime(p) - p = y
```

当x与y已知时，p*(q+x) = a和q*(p+y) = b构成二元二次方程组，可以通过爆破x与y尝试解方程，在正整数范围内求出解即为p和q之一。完整解题代码如下：

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import gmpy2
from libnum import n2s
n = 803086050719548165642433145523144313577352447653641953474510663716576290947829214155684689214655355560930191
4884176422322286739546193682236355823149096731058044933046552926707682168435727800175783373045726692093694148718
5216105905237188130968958835332453312446506758124065406949481212583948220229987732334006231621379493817721953513
3954897742256454605418891854238208847166679584218501900202508354316299173930993597270587194378773378449173550090
5013651061284020447578230135075211268405413254368439549259917312445348808412659422810647972872286215701325216318
64198549820234928137490589227989461283500918694414329876125 7
c = 330412463971933434999766363211057930667393277770584064857577467142742413428768098831412931259336108760624381
9528298610131797078262351307396831985397555390640151391138633431951746748156610463582479645561779194981806129898
0098765178994508408755696759767651556084467992036999274488350047567071512810448596766955333737557982738925031947
5394899794765310069084188092544505917549431419860547502393956775040990721765429143061510225852399839423143679690
2635077995829477347316754739938980814293304289318417443493019704073164585505217658570214989150175123757038125380
99605076157202198657393415547064109167866445108006571926120 7
a = 896150685275388363156021241540083002866369345996173348675090530766227153658093717400373165588717964339068444
6407099586929365408257788757819718240804517503533928508572800283822031406847467097522877846424008808433180742072
012136448676501116966974755339366165091211422822730857994016426987710197372845225287938 3
b = 896150685275388363156021241540083002866369345996173348675090530766227153658093717400373165588717964339068444
6407099586929365408257788757819718240804517278179870317365057473764491451559152225675884808995557871345871523453
666441521652683096783186230151863658670221218908795913650933410277285565766409157063007 9

for x in range(1, 1000):
    for y in range(1, 1000):
        delta = (a-b-x*y)**2 + 4*a*x*y
        tmp = gmpy2.iroot(delta, 2)
        if(tmp[1]):
            p = (a-b-x*y + int(tmp[0])) / (2*x)
            if (a*b)%p == 0:
                print p
                pp = gmpy2.next_prime(p)
                print pp
                tn = n // (p*pp)
                tq = int(gmpy2.iroot(tn,2)[0])
                qq = gmpy2.next_prime(tq)
                q = tn // qq
                print q
                print qq
                print n - p * pp * q * qq
                phi = (p-1) * (q-1) * (pp-1) * (qq-1)
                d = gmpy2.invert(0x10001, phi)
                print n2s(pow(c,d,n))
```

simpleRSA

根据代码，我们可以发现n很大，但是是三个数pqr的乘积，简单测试发现常规wiener脚本和boneh-durfee脚本均不可用，需要深入理解算法的原理。百度搜索wiener攻击，发现这样一种解释：

首先引入一个概念：连分数。对于任意两个正整数a、b，我们尝试构造这样一个分数的序列：

$$\frac{a_0}{b_0},\frac{a_1}{b_1},\ldots,\frac{a_n}{b_n}$$

使其不断接近a/b。存在这样一种构造方法，即首先将a/b完整展开为繁分数形式，然后每次计算其中的一部分逐渐逼近a/b。举一个简单的例子，例如a=11369, b=31337

$$\frac{a}{b}=\frac{11369}{31337}=\frac{1}{\frac{31337}{11369}}=\frac{1}{2+\frac{8599}{11369}}=\frac{1}{2+\frac{1}{\frac{11369}{8599}}}=\frac{1}{2+\frac{1}{1+\frac{2770}{8599}}}$$

$$=\frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{289}{2770}}}}=\frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9+\frac{169}{289}}}}}=\frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9+\frac{1}{1+\frac{120}{169}}}}}}$$

$$=\frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9+\frac{1}{1+\frac{1}{1+\frac{49}{120}}}}}}}=\frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9+\frac{1}{1+\frac{1}{1+\frac{1}{2+\frac{22}{49}}}}}}}}=\frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9+\frac{1}{1+\frac{1}{1+\frac{1}{2+\frac{1}{2+\frac{5}{22}}}}}}}}}$$

$=\cdots\cdots$

（Word公式编辑器只能写10层，这俩破数选的真不好。。）
我们将最后的分数部分去掉，生成这样一个数列：

$$\frac{1}{2},\ \frac{1}{2+\frac{1}{1}},\ \frac{1}{2+\frac{1}{1+\frac{1}{3}}},\ \frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9}}}},\ \frac{1}{2+\frac{1}{1+\frac{1}{3+\frac{1}{9+\frac{1}{1}}}}},\ \cdots$$

这就是一个逐渐逼近，并最终达到a/b的连分数序列。

接下来回到题目。题目满足基础等式

$$e*d=k*phi(n)+1$$

所以

$$\frac{e}{phi(n)}=\frac{k}{d}+\frac{1}{d*phi(n)}$$

来看一下各自的大小：pqr各1024位，所以n是3072位左右；phi(n)和e也差不多；d是512位；根据算式可知k也差不多，所以原式变为

$$\frac{3072bits}{3072bits}=\frac{512bits}{512bits}+\frac{1}{3584bits}$$

注意到phi(n)<n，所以

$$\frac{e}{phi(n)}>\frac{e}{n}$$

$$\frac{e}{phi(n)}-\frac{e}{n}=e*\frac{n-phi(n)}{phi(n)*n}=3072bits*\frac{2048bits}{6144bits}$$

二者差距是很小的，所以结论是e/n与k/d相差无几。

wiener算法的强大之处在于，wiener本人证明了在e/n的连分式展开中，必定有一个是k/d。本题我们也采用相同的方式展开，找到其中512位的那一项即为d，求即pow(c, d, n)解出明文。完整代码如下：

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from libnum import n2s

def continuedFraction(x, y):
    ret = []
    while y:
        ret.append(x / y)
        x, y = y, x % y
    return ret

def expand(ctnf):
    _ctnf = ctnf
    _ctnf.reverse()
    numerator = 0
    denominator = 1
    for x in _ctnf:
        numerator, denominator = denominator, x * denominator + numerator
    return (numerator, denominator)

def progressiveFraction(x, y):
    cfe = continuedFraction(x, y)
    cfeL = len(cfe)
    ret = []
    for i in xrange(1, cfeL):
        ret.append(expand(cfe[0 : i]))
    return ret

e = 107229542594413650703993867710144248121351940812514823388044284920635337968198930500057038709315223626320339
572697469295981931541078118009421620910006953079140749551088264078192056473221432789809994479271425362204787315 2
630438060151644601786843683746256407925709702163565141004356238879406385566586704226148537863811717298966607314 7
477375517243795166753766347714558839760690071342189824351701606478485494122891289820706478327744463450624893740 9
2673169618836701679
n = 182722199269284917924406983427381656571427650530524610343596288746152038170973992722305523995396518245125219
476893570262805658703417380060582742404328167318360647873618992737774557537990887645648501683241680602925497276 9
617393560238494326078940842295153029285394491783712384990125100774596477064482280829407856014835231711788990066 6
765344144147410677595641023316146667137970738112450995121305286004640994927346716890849900360778600422384549089 6
0841595107122933173
c = 107992917411082049405935541505910422990526876308915777137465793264671101748870153646068731964836254956331312
526806972241214802388562696264091585231729791642172581807781423729280721895257411114191815839119062136250886284 2
932945783059181952614317289116405878741758913351697905289993651105968169193211242144991434715552952340791545323 2
700657635298650103261928243346844132123577082752590962025090428380811500557276504438874382539646074149442458779 0
4002580997866300452

s = progressiveFraction(e, n)
for i in s:
    if len(bin(i[0]))-2 == 256:
        print n2s(pow(c,i[0],n))
```

easy matrix

（第二天没打，直接搬运）
解题方法是利用Babai's Nearest Plane算法计算CVP，再构造格解决
解题代码来源：Gm1y大佬的博客
祥云杯2020 Crypto wp

```
from sage.modules.free_module_integer import IntegerLattice

row = 128
```

```
prime = 2129
column=42

ma=[[133, 210, 141, 293, 30, 290, 445, 377, 292, 201, 92, 68, 374, 169, 69, 70, 195, 495, 32, 361, 202, 158, 198
, 187, 95, 449, 447, 507, 315, 475, 97, 271, 339, 123, 121, 279, 41, 341, 280, 325, 24, 293, 467, 19, 59, 284, 4
88, 349, 188, 373, 240, 115, 50, 173, 270, 82, 157, 398, 393, 376, 365, 228, 425, 502, 375, 85, 387, 133, 450, 5
01, 176, 477, 340, 221, 115, 233, 451, 202, 409, 484, 418, 268, 340, 345, 134, 363, 105, 168, 385, 135, 219, 415
, 488, 454, 85, 120, 215, 455, 365, 145, 249, 158, 399, 161, 344, 372, 293, 122, 275, 253, 128, 432, 326, 72, 51
0, 310, 137, 293, 88, 462, 471, 479, 411, 17, 141, 375, 345, 322], [22, 367, 369, 83, 326, 195, 405, 351, 235, 4
30, 198, 455, 495, 509, 151, 208, 171, 202, 103, 328, 449, 95, 20, 178, 288, 179, 9, 333, 60, 489, 298, 367, 326
, 441, 433, 195, 197, 330, 10, 63, 358, 336, 197, 238, 424, 167, 154, 157, 63, 53, 72, 67, 363, 349, 94, 197, 18
4, 463, 223, 34, 295, 392, 213, 11, 303, 295, 207, 399, 370, 429, 111, 93, 257, 245, 45, 56, 27, 68, 69, 32, 24,
 478, 289, 449, 82, 61, 397, 431, 103, 356, 8, 238, 316, 320, 169, 485, 368, 158, 142, 430, 22, 439, 100, 455, 3
74, 384, 449, 307, 38, 293, 215, 70, 335, 88, 94, 360, 364, 93, 178, 374, 51, 433, 447, 484, 450, 123, 313, 237]
, [100, 199, 503, 511, 131, 396, 425, 44, 473, 277, 42, 303, 355, 169, 61, 172, 103, 324, 57, 467, 195, 506, 186
, 429, 230, 498, 242, 143, 86, 288, 149, 268, 500, 210, 261, 135, 57, 210, 469, 281, 72, 308, 502, 168, 488, 372
, 49, 314, 368, 489, 475, 156, 224, 310, 219, 190, 189, 36, 388, 497, 197, 119, 19, 444, 292, 94, 9, 21, 24, 228
, 413, 3, 97, 439, 52, 123, 317, 61, 371, 266, 263, 44, 32, 13, 361, 324, 490, 235, 99, 236, 408, 340, 417, 324,
 299, 152, 159, 277, 124, 27, 508, 12, 268, 48, 15, 67, 47, 370, 201, 237, 81, 65, 475, 11, 291, 227, 425, 247,
365, 205, 364, 468, 511, 300, 491, 154, 404, 189], [73, 457, 149, 367, 38, 476, 160, 112, 400, 330, 370, 18, 239
, 363, 213, 378, 138, 21, 510, 240, 18, 378, 485, 204, 422, 287, 241, 95, 340, 138, 349, 150, 338, 15, 248, 205,
 97, 459, 92, 440, 66, 107, 124, 285, 36, 391, 12, 19, 38, 468, 374, 395, 453, 59, 136, 163, 336, 86, 386, 457,
107, 491, 70, 459, 113, 59, 432, 180, 164, 277, 456, 474, 16, 35, 272, 265, 115, 234, 418, 249, 78, 252, 135, 32
3, 31, 2, 486, 323, 174, 70, 443, 55, 90, 260, 231, 256, 486, 240, 284, 59, 388, 351, 430, 71, 317, 276, 93, 504
, 378, 226, 507, 258, 167, 386, 85, 478, 177, 103, 42, 176, 285, 181, 98, 120, 239, 400, 71, 215], [192, 277, 52
, 44, 432, 302, 58, 172, 292, 104, 482, 328, 143, 14, 1, 246, 272, 58, 331, 131, 226, 413, 168, 303, 429, 419, 4
53, 429, 457, 166, 248, 490, 470, 420, 97, 502, 387, 133, 58, 464, 272, 149, 438, 57, 70, 121, 86, 371, 304, 253
, 160, 192, 297, 311, 254, 244, 104, 168, 81, 237, 365, 108, 58, 130, 341, 170, 243, 129, 492, 394, 17, 354, 124
, 460, 124, 381, 212, 205, 141, 369, 144, 395, 32, 446, 451, 137, 458, 85, 230, 428, 364, 210, 409, 229, 41, 194
, 393, 508, 497, 302, 433, 80, 412, 5, 141, 380, 190, 191, 7, 290, 492, 382, 447, 25, 158, 431, 41, 264, 307, 48
8, 236, 411, 133, 160, 342, 83, 162, 436], [287, 413, 405, 325, 112, 378, 41, 43, 292, 195, 263, 70, 427, 243, 3
06, 288, 391, 352, 92, 78, 171, 12, 119, 252, 17, 394, 486, 411, 99, 422, 92, 408, 176, 476, 420, 286, 258, 399,
 439, 374, 312, 430, 118, 469, 264, 414, 341, 169, 84, 28, 10, 491, 197, 80, 111, 196, 263, 211, 198, 264, 112,
70, 34, 429, 54, 74, 316, 147, 336, 123, 81, 321, 136, 486, 277, 103, 104, 256, 92, 379, 400, 486, 295, 359, 65,
 457, 51, 242, 441, 69, 448, 158, 66, 422, 354, 208, 373, 135, 78, 402, 120, 7, 461, 508, 281, 496, 313, 195, 45
9, 504, 466, 289, 405, 1, 23, 341, 48, 175, 411, 58, 339, 209, 509, 240, 36, 199, 379, 248], [340, 187, 50, 85,
473, 355, 19, 133, 266, 243, 1, 290, 352, 141, 30, 303, 324, 19, 380, 360, 75, 83, 426, 435, 1, 60, 129, 408, 17
1, 52, 439, 160, 263, 65, 236, 294, 250, 371, 270, 61, 133, 407, 212, 14, 330, 467, 59, 258, 414, 163, 40, 122,
10, 370, 424, 487, 331, 261, 76, 375, 178, 91, 424, 273, 489, 20, 382, 306, 470, 317, 395, 287, 252, 232, 430, 2
74, 143, 146, 242, 96, 289, 53, 312, 353, 250, 268, 180, 446, 228, 199, 508, 280, 491, 238, 124, 230, 234, 286,
180, 153, 336, 26, 316, 356, 96, 40, 286, 136, 408, 440, 266, 181, 223, 473, 341, 4, 494, 38, 172, 258, 154, 306
, 34, 94, 368, 236, 452, 270], [483, 13, 49, 109, 226, 479, 40, 333, 275, 290, 110, 23, 406, 302, 178, 389, 126,
 324, 221, 89, 476, 246, 98, 22, 456, 235, 8, 22, 11, 509, 8, 493, 181, 210, 196, 128, 432, 245, 255, 85, 317, 1
47, 91, 17, 54, 464, 498, 383, 445, 191, 461, 307, 311, 456, 508, 211, 491, 362, 88, 283, 24, 328, 143, 238, 11,
 485, 108, 335, 231, 98, 391, 166, 80, 1, 174, 26, 492, 141, 60, 323, 197, 204, 262, 16, 8, 474, 501, 33, 374, 2
11, 50, 61, 384, 146, 372, 161, 228, 314, 181, 171, 16, 478, 357, 145, 118, 467, 363, 410, 82, 498, 215, 410, 91
, 481, 402, 215, 142, 283, 374, 405, 104, 262, 171, 92, 384, 479, 22, 328], [298, 281, 454, 291, 338, 412, 372,
461, 49, 43, 497, 428, 499, 148, 244, 207, 79, 26, 216, 470, 197, 66, 286, 211, 356, 77, 276, 110, 287, 66, 86,
380, 504, 84, 372, 475, 340, 26, 339, 437, 89, 182, 491, 149, 331, 500, 371, 262, 339, 304, 453, 362, 101, 36, 2
70, 288, 288, 473, 107, 313, 363, 143, 475, 55, 167, 56, 33, 152, 274, 212, 309, 89, 162, 226, 274, 429, 324, 16
9, 371, 99, 146, 202, 177, 381, 215, 74, 120, 93, 488, 11, 114, 61, 358, 90, 221, 318, 452, 454, 35, 202, 127, 2
34, 129, 233, 428, 502, 417, 472, 245, 491, 396, 64, 277, 259, 114, 30, 455, 271, 183, 358, 171, 309, 366, 88, 3
5, 112, 94, 135], [268, 179, 137, 341, 379, 113, 338, 78, 112, 394, 6, 336, 244, 241, 425, 438, 89, 87, 493, 171
, 445, 399, 269, 18, 506, 252, 310, 452, 499, 361, 252, 343, 138, 240, 246, 477, 99, 463, 57, 153, 482, 72, 22,
318, 504, 20, 374, 62, 119, 370, 23, 369, 172, 228, 91, 289, 19, 405, 183, 152, 281, 26, 151, 188, 207, 81, 9, 6
8, 249, 199, 234, 316, 19, 70, 157, 317, 286, 310, 93, 426, 57, 149, 34, 156, 279, 407, 131, 359, 86, 430, 272,
169, 387, 102, 489, 227, 467, 445, 382, 118, 91, 42, 120, 328, 505, 0, 181, 451, 225, 216, 303, 424, 32, 141, 18
0, 315, 145, 287, 499, 111, 315, 411, 416, 380, 126, 228, 346, 10], [124, 480, 127, 192, 138, 285, 126, 327, 230
, 85, 171, 295, 461, 195, 140, 289, 386, 59, 261, 453, 403, 194, 12, 511, 95, 122, 193, 213, 265, 70, 55, 372, 4
```

28, 392, 409, 219, 476, 113, 264, 279, 475, 259, 297, 458, 345, 49, 161, 314, 312, 49, 130, 48, 16, 240, 502, 29
4, 186, 276, 267, 290, 278, 107, 18, 453, 213, 69, 209, 214, 52, 175, 142, 236, 164, 489, 334, 125, 420, 195, 16
, 16, 185, 333, 478, 305, 1, 504, 240, 248, 36, 411, 479, 260, 222, 422, 116, 273, 459, 165, 59, 164, 35, 220, 4
46, 131, 492, 485, 326, 34, 297, 62, 257, 56, 64, 190, 302, 434, 402, 80, 224, 294, 226, 180, 243, 510, 299, 329
, 78, 293], [254, 411, 419, 227, 475, 360, 107, 381, 346, 48, 67, 53, 45, 426, 322, 6, 102, 179, 403, 443, 394,
455, 102, 406, 2, 289, 228, 417, 133, 66, 211, 338, 96, 65, 343, 157, 244, 432, 71, 6, 377, 6, 10, 473, 146, 20,
367, 0, 204, 391, 46, 321, 410, 501, 298, 167, 231, 328, 414, 422, 460, 473, 320, 143, 435, 433, 479, 316, 46,
472, 62, 441, 332, 149, 92, 309, 159, 151, 469, 150, 482, 29, 342, 344, 376, 363, 73, 64, 353, 141, 263, 336, 36
7, 248, 22, 90, 487, 83, 261, 273, 425, 297, 432, 200, 15, 308, 380, 372, 177, 134, 507, 204, 307, 493, 257, 195
, 252, 400, 280, 110, 240, 141, 83, 269, 108, 81, 195, 249], [173, 160, 489, 196, 177, 147, 289, 131, 68, 197, 2
21, 468, 436, 334, 84, 242, 458, 490, 388, 304, 167, 85, 323, 248, 264, 170, 186, 456, 482, 376, 101, 191, 345,
184, 482, 477, 142, 112, 145, 105, 215, 267, 338, 283, 7, 244, 126, 404, 153, 98, 188, 285, 491, 323, 91, 173, 4
28, 253, 69, 284, 138, 311, 509, 420, 434, 313, 16, 287, 460, 215, 364, 364, 180, 425, 203, 121, 267, 198, 308,
399, 503, 250, 204, 479, 351, 256, 35, 94, 227, 391, 20, 261, 262, 491, 122, 99, 75, 54, 467, 352, 348, 94, 285,
296, 221, 446, 7, 194, 29, 280, 500, 266, 253, 247, 84, 398, 197, 79, 76, 65, 306, 97, 25, 268, 483, 280, 148,
88], [352, 205, 226, 154, 91, 83, 147, 425, 452, 23, 498, 257, 292, 426, 438, 472, 151, 343, 129, 92, 192, 358,
366, 62, 10, 313, 382, 280, 83, 244, 77, 214, 402, 167, 457, 275, 436, 39, 160, 114, 338, 192, 283, 402, 477, 25
4, 89, 355, 412, 472, 270, 504, 351, 452, 296, 129, 259, 266, 376, 296, 54, 436, 347, 458, 434, 189, 400, 107, 4
34, 503, 133, 382, 204, 477, 25, 30, 352, 436, 248, 8, 470, 328, 335, 281, 450, 359, 191, 364, 85, 475, 497, 170
, 100, 442, 102, 358, 325, 165, 128, 264, 457, 215, 245, 287, 71, 253, 354, 185, 238, 132, 244, 321, 125, 412, 1
60, 414, 484, 368, 352, 65, 410, 502, 172, 300, 417, 396, 510, 416], [134, 213, 20, 456, 215, 143, 403, 478, 25,
249, 187, 281, 386, 122, 113, 210, 93, 371, 18, 357, 323, 428, 401, 19, 31, 432, 87, 15, 28, 171, 413, 120, 339
, 434, 123, 281, 407, 165, 342, 464, 480, 206, 380, 322, 11, 228, 62, 251, 7, 149, 496, 75, 325, 90, 267, 308, 3
61, 139, 349, 199, 355, 508, 490, 465, 83, 110, 133, 372, 181, 229, 285, 375, 224, 417, 43, 241, 424, 483, 383,
319, 488, 100, 271, 206, 98, 213, 224, 405, 469, 132, 343, 149, 159, 504, 191, 175, 311, 350, 195, 323, 368, 233
, 12, 357, 136, 327, 338, 133, 45, 257, 119, 315, 319, 111, 108, 416, 477, 488, 254, 407, 329, 302, 104, 155, 26
4, 296, 86, 154], [359, 345, 463, 387, 261, 434, 425, 167, 429, 288, 24, 250, 110, 427, 129, 66, 13, 180, 108, 1
62, 361, 79, 121, 32, 284, 461, 476, 188, 48, 233, 238, 223, 287, 354, 329, 6, 199, 491, 6, 53, 237, 456, 346, 4
86, 115, 27, 316, 53, 115, 434, 103, 274, 38, 90, 49, 190, 266, 392, 499, 113, 289, 223, 180, 376, 10, 460, 145,
393, 490, 173, 410, 469, 411, 455, 196, 234, 355, 92, 349, 133, 459, 465, 267, 188, 163, 158, 129, 100, 151, 35
5, 413, 18, 146, 226, 184, 310, 40, 340, 389, 206, 231, 31, 104, 275, 138, 195, 113, 304, 321, 42, 247, 66, 239,
255, 398, 360, 184, 441, 108, 206, 151, 370, 363, 112, 162, 125, 481, 106], [271, 450, 447, 370, 422, 112, 232,
81, 290, 329, 413, 10, 150, 62, 179, 60, 414, 449, 326, 126, 258, 332, 125, 59, 119, 226, 153, 182, 130, 402, 3
03, 70, 121, 291, 217, 10, 60, 347, 60, 403, 301, 235, 280, 226, 377, 285, 405, 292, 147, 343, 308, 455, 350, 14
3, 312, 443, 495, 477, 410, 438, 166, 219, 381, 187, 505, 13, 35, 266, 415, 474, 44, 396, 225, 147, 324, 73, 298
, 118, 127, 219, 281, 476, 107, 390, 167, 431, 358, 441, 247, 9, 14, 76, 205, 3, 504, 390, 59, 495, 511, 58, 186
, 446, 354, 92, 114, 121, 303, 287, 272, 219, 412, 336, 416, 343, 383, 356, 391, 79, 274, 363, 76, 472, 458, 358
, 142, 52, 165, 469], [219, 210, 274, 470, 98, 315, 352, 328, 161, 501, 361, 284, 348, 509, 474, 397, 276, 212,
0, 200, 419, 50, 274, 446, 45, 86, 482, 166, 149, 125, 236, 292, 209, 163, 101, 239, 391, 53, 462, 250, 167, 348
, 71, 142, 323, 120, 98, 83, 430, 119, 167, 195, 263, 185, 28, 466, 263, 416, 238, 209, 0, 462, 115, 276, 2, 46,
29, 366, 174, 484, 72, 122, 449, 421, 167, 453, 219, 221, 309, 335, 145, 444, 470, 392, 342, 85, 190, 221, 104,
463, 48, 368, 213, 183, 421, 59, 244, 140, 155, 53, 122, 68, 23, 267, 123, 201, 334, 87, 145, 402, 233, 399, 28
1, 95, 330, 181, 115, 134, 290, 361, 48, 362, 14, 65, 405, 84, 50, 343], [187, 122, 304, 228, 456, 322, 482, 300
, 424, 466, 19, 31, 287, 376, 211, 451, 452, 312, 376, 10, 417, 322, 425, 160, 298, 267, 404, 327, 451, 180, 64,
433, 287, 404, 289, 207, 133, 99, 251, 23, 126, 86, 350, 137, 439, 141, 186, 478, 329, 392, 237, 149, 485, 251,
57, 493, 153, 10, 451, 208, 454, 363, 59, 14, 252, 9, 297, 349, 142, 497, 189, 48, 184, 68, 10, 423, 246, 427,
500, 379, 238, 31, 291, 193, 129, 166, 201, 69, 398, 120, 323, 60, 475, 146, 276, 219, 42, 155, 270, 11, 295, 25
, 276, 397, 81, 32, 322, 350, 208, 496, 244, 192, 324, 376, 403, 496, 18, 8, 221, 107, 326, 378, 442, 258, 102,
466, 234, 363], [469, 238, 93, 13, 477, 186, 471, 175, 161, 399, 7, 71, 35, 386, 209, 145, 508, 421, 292, 26, 11
4, 125, 79, 380, 262, 64, 267, 313, 155, 55, 139, 262, 168, 375, 157, 185, 360, 352, 429, 242, 506, 444, 27, 144
, 451, 324, 503, 91, 421, 79, 6, 201, 484, 404, 298, 52, 70, 339, 504, 101, 3, 353, 448, 32, 277, 481, 490, 471,
385, 316, 263, 132, 351, 144, 356, 167, 225, 8, 266, 357, 242, 67, 42, 252, 343, 160, 425, 138, 92, 251, 354, 1
22, 137, 240, 282, 400, 287, 292, 37, 88, 267, 218, 354, 325, 106, 90, 372, 337, 417, 506, 263, 104, 311, 283, 2
98, 442, 56, 408, 426, 172, 161, 251, 161, 361, 434, 104, 181, 194], [113, 81, 318, 2, 390, 373, 198, 208, 500,
34, 131, 337, 25, 131, 76, 419, 493, 174, 136, 147, 14, 64, 485, 181, 27, 287, 7, 352, 421, 17, 381, 123, 340, 3
88, 54, 470, 451, 57, 407, 28, 204, 16, 34, 156, 418, 396, 503, 29, 98, 223, 472, 65, 160, 318, 205, 364, 25, 19
2, 128, 126, 264, 96, 214, 368, 255, 95, 254, 441, 285, 347, 161, 283, 414, 387, 419, 240, 240, 466, 114, 391, 1
29, 160, 107, 129, 460, 82, 283, 371, 26, 188, 125, 452, 158, 358, 492, 273, 83, 490, 415, 288, 211, 447, 26, 41
9, 232, 99, 146, 400, 308, 242, 172, 246, 225, 216, 468, 411, 411, 23, 342, 304, 483, 103, 174, 491, 268, 204, 1
34, 159], [187, 169, 103, 244, 485, 337, 424, 334, 178, 346, 451, 401, 36, 70, 279, 482, 471, 391, 114, 353, 362

, 45, 87, 318, 373, 201, 442, 449, 273, 497, 254, 169, 250, 228, 209, 161, 210, 403, 259, 201, 452, 394, 298, 30
0, 262, 223, 215, 460, 9, 447, 498, 315, 51, 53, 505, 2, 271, 414, 154, 12, 198, 305, 213, 71, 308, 107, 110, 40
9, 396, 453, 451, 320, 280, 1, 386, 355, 332, 59, 75, 384, 164, 71, 166, 260, 38, 88, 401, 377, 296, 35, 4, 463,
136, 323, 257, 246, 159, 505, 432, 18, 123, 326, 75, 326, 452, 152, 466, 16, 462, 443, 146, 179, 416, 108, 55,
84, 393, 151, 159, 110, 3, 202, 262, 300, 173, 445, 277, 395], [325, 375, 135, 339, 380, 295, 457, 380, 152, 284
, 480, 120, 164, 75, 262, 23, 12, 197, 374, 364, 451, 219, 20, 78, 120, 305, 387, 377, 350, 41, 218, 166, 165, 2
39, 167, 131, 281, 510, 479, 1, 88, 11, 74, 123, 163, 113, 495, 299, 339, 283, 116, 185, 282, 463, 509, 289, 104
, 106, 136, 506, 488, 236, 198, 394, 192, 438, 376, 349, 0, 80, 289, 138, 155, 106, 494, 327, 298, 16, 140, 177,
380, 101, 395, 332, 474, 140, 114, 258, 216, 252, 237, 339, 252, 48, 361, 397, 104, 263, 313, 343, 366, 157, 18
, 18, 190, 423, 485, 210, 263, 428, 399, 170, 17, 37, 389, 348, 42, 433, 417, 49, 137, 458, 356, 503, 193, 107,
173, 242], [155, 422, 293, 482, 451, 490, 40, 241, 72, 419, 473, 271, 51, 13, 292, 37, 448, 501, 503, 174, 353,
144, 40, 179, 24, 405, 288, 81, 489, 394, 153, 32, 487, 54, 393, 289, 362, 3, 303, 305, 283, 159, 216, 482, 373,
120, 147, 10, 349, 405, 385, 101, 308, 189, 200, 464, 127, 414, 130, 317, 333, 369, 44, 308, 71, 253, 65, 76, 4
13, 495, 78, 323, 303, 77, 229, 432, 186, 285, 411, 434, 145, 0, 26, 199, 355, 132, 364, 202, 406, 42, 379, 47,
121, 5, 157, 219, 47, 294, 362, 238, 299, 192, 12, 228, 71, 233, 315, 474, 276, 79, 431, 193, 511, 18, 430, 138,
439, 113, 271, 293, 265, 250, 365, 374, 327, 370, 156, 371], [486, 154, 142, 140, 423, 24, 66, 354, 333, 378, 3
17, 131, 22, 35, 64, 9, 55, 210, 123, 511, 59, 208, 395, 366, 492, 33, 205, 154, 348, 35, 49, 298, 461, 213, 186
, 325, 245, 460, 358, 74, 509, 496, 38, 104, 415, 341, 60, 213, 40, 166, 400, 435, 505, 372, 504, 240, 57, 180,
291, 143, 348, 445, 386, 314, 388, 179, 249, 420, 463, 379, 510, 200, 377, 459, 286, 381, 287, 336, 51, 336, 282
, 397, 93, 506, 104, 182, 338, 384, 155, 417, 268, 100, 195, 235, 59, 123, 167, 254, 332, 323, 400, 46, 427, 94,
497, 124, 267, 75, 345, 417, 271, 386, 347, 184, 248, 54, 311, 315, 236, 348, 113, 71, 350, 376, 502, 253, 163,
397], [63, 60, 158, 328, 351, 134, 181, 39, 462, 429, 289, 472, 368, 396, 333, 391, 370, 348, 327, 89, 462, 320
, 217, 427, 276, 47, 260, 112, 63, 336, 49, 446, 86, 231, 318, 442, 476, 169, 196, 190, 238, 26, 112, 292, 385,
406, 346, 434, 12, 20, 120, 229, 160, 259, 21, 233, 79, 179, 270, 322, 267, 426, 20, 375, 283, 351, 354, 280, 37
1, 474, 231, 99, 379, 511, 326, 334, 119, 362, 494, 182, 143, 107, 386, 95, 449, 165, 325, 476, 284, 191, 286, 4
67, 139, 357, 55, 129, 181, 126, 243, 52, 349, 387, 179, 436, 114, 367, 36, 210, 221, 223, 244, 451, 29, 424, 31
1, 103, 238, 147, 4, 184, 66, 317, 164, 420, 218, 211, 28, 256], [354, 51, 338, 228, 156, 308, 266, 278, 410, 12
5, 433, 162, 207, 210, 0, 392, 324, 60, 176, 376, 69, 36, 186, 249, 405, 325, 51, 442, 35, 508, 238, 391, 165, 6
9, 394, 421, 353, 389, 61, 158, 488, 409, 20, 503, 53, 209, 364, 227, 274, 156, 363, 235, 154, 2, 223, 332, 233,
366, 471, 121, 172, 174, 77, 233, 282, 302, 482, 253, 410, 100, 35, 324, 493, 376, 416, 473, 290, 173, 14, 129,
25, 192, 225, 163, 250, 320, 80, 119, 302, 250, 122, 390, 245, 365, 361, 318, 481, 118, 180, 365, 129, 429, 329
, 365, 396, 11, 369, 409, 96, 472, 488, 135, 122, 175, 122, 293, 359, 290, 137, 304, 225, 396, 136, 275, 501, 19
1, 105, 383], [422, 296, 383, 132, 88, 62, 249, 166, 32, 505, 257, 308, 11, 496, 99, 335, 195, 39, 506, 21, 46,
360, 480, 286, 20, 44, 359, 142, 309, 172, 69, 460, 483, 318, 374, 103, 59, 460, 91, 7, 87, 32, 391, 287, 170, 2
30, 141, 154, 96, 219, 120, 483, 118, 497, 195, 493, 356, 131, 133, 253, 290, 481, 146, 247, 105, 210, 455, 420,
474, 141, 362, 251, 340, 442, 107, 143, 435, 62, 505, 378, 250, 233, 351, 303, 434, 35, 99, 406, 315, 467, 427,
476, 496, 83, 149, 215, 408, 157, 105, 140, 292, 199, 446, 15, 458, 417, 220, 112, 470, 416, 229, 390, 467, 250
, 122, 357, 353, 127, 89, 174, 116, 231, 237, 207, 393, 156, 38, 19], [190, 511, 246, 352, 61, 240, 26, 480, 464
, 346, 303, 386, 396, 18, 483, 420, 463, 10, 136, 326, 45, 328, 146, 478, 96, 382, 218, 405, 107, 125, 274, 265,
118, 428, 464, 246, 407, 431, 24, 170, 250, 194, 139, 372, 158, 91, 451, 384, 124, 179, 261, 393, 214, 274, 496
, 263, 76, 465, 165, 203, 63, 368, 257, 305, 278, 238, 181, 460, 127, 4, 403, 309, 392, 34, 17, 319, 81, 312, 27
3, 8, 345, 186, 368, 353, 116, 74, 360, 89, 418, 61, 78, 453, 311, 177, 356, 59, 415, 157, 138, 354, 146, 480, 5
1, 155, 356, 119, 473, 400, 269, 462, 413, 410, 481, 508, 377, 192, 137, 177, 415, 287, 122, 415, 191, 367, 472,
288, 446, 158], [358, 329, 327, 261, 392, 389, 61, 90, 510, 235, 257, 395, 444, 476, 108, 95, 253, 78, 354, 62,
47, 197, 68, 222, 359, 95, 207, 335, 0, 108, 277, 430, 1, 261, 196, 210, 406, 235, 54, 6, 227, 117, 105, 181, 2
6, 170, 160, 485, 285, 32, 245, 152, 12, 378, 91, 33, 314, 158, 381, 419, 390, 119, 361, 303, 1, 430, 510, 211,
339, 291, 326, 192, 388, 309, 224, 198, 455, 382, 208, 487, 448, 480, 154, 309, 498, 118, 445, 279, 507, 152, 52
, 391, 430, 453, 85, 102, 52, 41, 457, 255, 440, 383, 410, 504, 50, 149, 174, 12, 242, 224, 181, 362, 429, 329,
99, 47, 98, 377, 426, 141, 18, 419, 465, 72, 92, 25, 308, 181], [185, 278, 19, 368, 235, 72, 6, 126, 205, 432, 1
7, 147, 194, 411, 376, 486, 363, 384, 344, 508, 196, 495, 444, 373, 285, 13, 478, 450, 264, 366, 408, 14, 368, 3
81, 395, 15, 297, 391, 300, 494, 166, 2, 152, 158, 438, 300, 267, 148, 429, 488, 238, 272, 237, 223, 189, 295, 2
03, 510, 355, 473, 231, 232, 56, 408, 69, 358, 484, 359, 260, 136, 443, 105, 267, 211, 161, 13, 354, 455, 389, 2
05, 186, 436, 128, 89, 27, 91, 138, 459, 201, 25, 68, 15, 252, 257, 278, 140, 457, 38, 271, 432, 473, 408, 38, 2
24, 26, 455, 142, 76, 18, 114, 128, 36, 291, 354, 260, 340, 9, 9, 96, 478, 146, 490, 389, 135, 40, 359, 413, 319
], [122, 233, 305, 191, 270, 389, 63, 235, 85, 244, 2, 320, 212, 216, 426, 318, 92, 89, 492, 266, 338, 423, 39,
271, 429, 298, 483, 310, 310, 268, 306, 276, 104, 473, 47, 198, 112, 111, 499, 250, 52, 263, 354, 465, 180, 83,
129, 219, 341, 158, 105, 132, 250, 139, 175, 442, 314, 94, 257, 329, 159, 200, 475, 207, 486, 314, 37, 115, 308,
405, 56, 253, 193, 283, 267, 51, 325, 404, 194, 286, 312, 324, 215, 122, 224, 487, 80, 292, 452, 284, 371, 171,
398, 5, 214, 106, 92, 294, 288, 161, 372, 202, 196, 241, 46, 294, 267, 94, 122, 293, 67, 88, 509, 160, 97, 309,

481, 357, 351, 250, 271, 261, 491, 291, 25, 190, 358, 78], [426, 340, 124, 4, 499, 28, 44, 56, 96, 132, 213, 20
1, 467, 284, 150, 422, 349, 159, 483, 65, 70, 502, 154, 11, 174, 296, 312, 159, 284, 335, 499, 302, 42, 33, 300,
84, 82, 394, 170, 93, 296, 472, 226, 224, 248, 39, 334, 171, 293, 36, 140, 74, 221, 225, 242, 508, 36, 319, 83,
167, 219, 58, 493, 332, 417, 486, 353, 23, 371, 347, 410, 151, 492, 190, 450, 154, 265, 319, 350, 495, 57, 194,
90, 97, 96, 416, 468, 102, 313, 487, 469, 129, 419, 235, 396, 510, 369, 483, 364, 432, 85, 364, 349, 424, 507,
120, 377, 246, 377, 107, 459, 244, 177, 142, 362, 419, 282, 292, 251, 488, 422, 81, 96, 429, 407, 55, 459, 204],
[357, 206, 338, 42, 420, 409, 220, 235, 203, 49, 251, 249, 67, 32, 292, 196, 244, 6, 226, 311, 187, 230, 446, 2
49, 63, 80, 318, 195, 455, 354, 150, 213, 292, 209, 276, 114, 68, 336, 99, 225, 147, 56, 228, 239, 463, 433, 336
, 440, 402, 51, 426, 301, 174, 32, 357, 368, 339, 342, 5, 137, 439, 472, 503, 325, 59, 125, 57, 465, 260, 94, 22
2, 112, 141, 464, 308, 397, 237, 425, 465, 3, 325, 187, 5, 319, 372, 101, 2, 49, 226, 107, 208, 9, 138, 481, 53,
379, 492, 182, 270, 57, 461, 506, 109, 217, 32, 45, 504, 13, 74, 306, 162, 362, 59, 436, 125, 27, 360, 357, 333
, 413, 314, 400, 347, 205, 491, 269, 322, 60], [332, 106, 1, 493, 482, 101, 318, 213, 343, 351, 193, 125, 284, 5
07, 218, 27, 489, 232, 36, 180, 381, 463, 498, 414, 444, 268, 193, 7, 372, 168, 255, 110, 340, 505, 461, 445, 50
3, 30, 246, 368, 390, 118, 0, 263, 459, 230, 11, 250, 440, 433, 162, 464, 46, 409, 266, 452, 435, 386, 437, 111,
510, 292, 137, 495, 83, 16, 181, 42, 173, 272, 389, 440, 295, 99, 456, 249, 60, 227, 470, 455, 276, 85, 496, 13
3, 221, 436, 307, 98, 88, 294, 63, 155, 318, 347, 464, 490, 106, 209, 400, 249, 204, 419, 378, 220, 47, 18, 164,
6, 63, 214, 84, 108, 88, 301, 309, 392, 42, 367, 150, 94, 203, 55, 505, 391, 294, 403, 163, 442], [289, 401, 43
3, 142, 358, 511, 69, 386, 441, 119, 218, 473, 495, 130, 399, 330, 422, 346, 14, 493, 109, 187, 21, 234, 222, 23
3, 354, 369, 101, 328, 264, 110, 97, 98, 34, 234, 349, 270, 479, 331, 499, 237, 83, 290, 104, 353, 133, 171, 13,
88, 49, 461, 235, 147, 15, 329, 29, 121, 473, 84, 315, 496, 239, 413, 233, 92, 362, 509, 357, 84, 320, 53, 96,
206, 154, 307, 409, 88, 139, 326, 349, 129, 494, 79, 339, 179, 231, 249, 416, 132, 462, 421, 290, 33, 147, 7, 46
9, 419, 465, 396, 467, 395, 508, 102, 234, 302, 488, 499, 204, 270, 240, 425, 440, 35, 345, 493, 120, 381, 434,
190, 424, 5, 353, 469, 15, 84, 14, 438], [291, 501, 380, 473, 8, 303, 295, 394, 439, 432, 406, 324, 76, 279, 201
, 139, 54, 245, 94, 292, 57, 191, 192, 335, 362, 439, 175, 320, 193, 28, 449, 472, 186, 506, 498, 276, 257, 2, 2
72, 209, 387, 444, 287, 114, 301, 460, 511, 62, 123, 4, 373, 395, 138, 181, 254, 19, 20, 153, 127, 158, 208, 278
, 151, 170, 368, 94, 402, 68, 480, 315, 340, 21, 247, 98, 262, 65, 76, 332, 490, 138, 229, 23, 410, 423, 403, 99
, 16, 487, 192, 178, 291, 462, 201, 329, 379, 365, 15, 94, 227, 439, 414, 216, 45, 152, 323, 460, 288, 233, 277,
227, 298, 336, 138, 91, 248, 72, 241, 385, 383, 505, 30, 484, 451, 91, 97, 470, 62, 230], [108, 342, 506, 57, 3
56, 56, 6, 323, 327, 332, 441, 259, 475, 501, 413, 105, 112, 229, 505, 365, 473, 226, 425, 405, 46, 489, 47, 431
, 306, 45, 112, 219, 505, 118, 45, 279, 480, 446, 125, 438, 156, 61, 97, 290, 460, 115, 172, 171, 308, 48, 217,
138, 42, 446, 357, 14, 487, 449, 403, 59, 253, 292, 187, 219, 100, 336, 391, 191, 172, 390, 290, 272, 338, 263,
176, 190, 282, 368, 188, 500, 280, 22, 33, 12, 97, 141, 497, 446, 314, 307, 232, 454, 6, 117, 490, 142, 232, 188
, 472, 119, 316, 362, 394, 119, 179, 353, 334, 385, 226, 183, 73, 126, 320, 369, 139, 470, 318, 413, 123, 222, 4
42, 173, 264, 256, 227, 403, 415, 69], [353, 425, 500, 233, 324, 53, 484, 175, 59, 134, 79, 187, 31, 294, 41, 94
, 487, 131, 460, 43, 253, 423, 233, 369, 135, 263, 351, 399, 201, 166, 491, 465, 396, 289, 140, 160, 218, 191, 3
77, 459, 482, 134, 299, 192, 85, 384, 177, 110, 197, 177, 43, 354, 388, 292, 297, 423, 332, 475, 85, 217, 150, 4
57, 58, 94, 87, 226, 130, 334, 425, 419, 488, 95, 294, 246, 140, 22, 301, 99, 329, 23, 53, 342, 331, 294, 282, 8
5, 99, 434, 110, 284, 441, 295, 42, 498, 66, 498, 386, 344, 377, 463, 372, 84, 453, 386, 42, 386, 111, 27, 187,
64, 91, 338, 201, 198, 26, 469, 427, 6, 388, 213, 441, 90, 125, 40, 257, 105, 324, 463], [464, 63, 92, 482, 226,
148, 502, 4, 508, 457, 61, 419, 344, 114, 306, 127, 254, 228, 491, 225, 4, 127, 161, 136, 290, 258, 487, 18, 25
4, 412, 7, 213, 456, 464, 167, 340, 14, 225, 342, 323, 251, 111, 0, 211, 344, 307, 340, 259, 320, 83, 197, 132,
229, 231, 186, 137, 394, 72, 223, 362, 197, 270, 135, 480, 93, 509, 75, 309, 210, 48, 343, 132, 230, 471, 439, 1
73, 497, 282, 341, 67, 310, 268, 34, 325, 321, 219, 129, 54, 13, 397, 471, 504, 419, 94, 242, 15, 65, 2, 480, 44
7, 20, 195, 160, 13, 173, 387, 488, 151, 115, 314, 110, 498, 105, 385, 291, 34, 500, 283, 27, 279, 89, 464, 80,
231, 390, 179, 80, 144], [197, 425, 464, 316, 93, 495, 356, 73, 369, 255, 246, 231, 35, 51, 202, 65, 266, 308, 5
2, 71, 410, 332, 85, 502, 100, 496, 496, 443, 6, 242, 164, 170, 414, 79, 272, 308, 249, 284, 298, 213, 127, 464,
195, 382, 377, 141, 56, 389, 382, 438, 285, 21, 478, 141, 155, 130, 396, 213, 396, 362, 160, 256, 494, 374, 371
, 356, 287, 410, 438, 296, 44, 68, 223, 167, 288, 87, 41, 288, 153, 69, 66, 407, 494, 369, 451, 163, 194, 323, 1
37, 33, 57, 214, 265, 196, 14, 191, 264, 79, 430, 51, 7, 505, 303, 69, 351, 461, 292, 379, 334, 167, 125, 31, 27
9, 54, 16, 306, 166, 94, 292, 226, 171, 288, 393, 151, 120, 163, 130, 466], [426, 168, 76, 144, 450, 201, 454, 7
2, 339, 416, 442, 60, 428, 120, 61, 507, 72, 160, 306, 313, 288, 70, 348, 396, 418, 113, 92, 244, 84, 343, 402,
171, 360, 438, 309, 478, 84, 411, 345, 106, 204, 137, 51, 7, 227, 147, 265, 392, 395, 158, 502, 123, 480, 52, 47
7, 142, 252, 408, 405, 196, 137, 256, 429, 510, 511, 393, 26, 348, 495, 169, 46, 143, 353, 348, 249, 312, 103, 4
83, 384, 332, 431, 133, 222, 275, 161, 10, 116, 71, 383, 173, 169, 30, 26, 351, 399, 64, 242, 247, 317, 187, 81,
147, 269, 511, 58, 42, 130, 368, 255, 478, 119, 462, 457, 62, 415, 382, 64, 275, 493, 489, 486, 306, 414, 380,
384, 217, 347, 490]]
res=[2087, 1418, 498, 2090, 539, 424, 1452, 61, 1447, 334, 963, 389, 1875, 514, 644, 977, 1473, 2062, 2082, 1501
, 1087, 1948, 674, 2026, 1867, 1065, 1413, 1913, 525, 1486, 793, 54, 569, 474, 239, 1237, 713, 1881, 937, 961, 1
539, 1252, 1766, 614, 1786, 1675, 2008, 1713, 1126, 637, 312, 241, 239, 1144, 118, 1451, 1905, 826, 226, 457, 43
7, 762, 1882, 1581, 1734, 1028, 1935, 1709, 23, 310, 496, 1395, 1248, 694, 1180, 1651, 108, 1, 13, 1386, 1300, 1

```
29, 525, 2127, 1845, 2026, 1972, 1099, 600, 2080, 1428, 452, 132, 158, 146, 267, 10, 1372, 939, 881, 2110, 2077,
 1519, 1926, 200, 502, 2006, 1577, 1636, 1024, 1887, 116, 43, 1085, 945, 71, 438, 936, 1235, 72, 1646, 581, 1654
, 635, 1977, 897, 1902, 29]

W=matrix(ZZ,ma)
cc= vector(ZZ,res)

# Babai's Nearest Plane algorithm
def Babai_closest_vector(M, G, target):
    small = target
    for _ in xrange(5):
        for i in reversed(range(M.nrows())):
            c = ((small * G[i]) / (G[i] * G[i])).round()
            small -=  M[i] * c
    return target - small

A1=matrix.identity(42)
Ap=matrix.identity(128)*2129
B=block_matrix([[Ap],[W]])
lattice = IntegerLattice(B, lll_reduce=True)
print("LLL done")
gram = lattice.reduced_basis.gram_schmidt()[0]
target = vector(ZZ, res)
re = Babai_closest_vector(lattice.reduced_basis, gram, target)
print("Closest Vector: {}".format(re))

R = IntegerModRing(prime)
M = Matrix(R, ma)
M=M.transpose()

ingredients = M.solve_right(re)
print("Ingredients: {}".format(ingredients))

m=''
for i in range(len(ingredients)):
    m+=chr(ingredients[i])
print m
```

blowfishgame

（第二天没打，直接搬运）

WriteUp by Nu1L

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from pwn import *
import base64
import re
from hashlib import sha384
from itertools import product

LOCAL = 0
VERBOSE = 0
if VERBOSE:
    context.log_level = 'debug'
if LOCAL:
    io = process(['python2', 'blowfishgame.py'])
else:
```

```
    io = remote('8.131.69.237', 15846)

table = string.ascii_letters + string.digits

# passPoW
io.recvuntil('sha384')
rec = io.recvline().decode()
suffix = re.findall(r'\(XXX\+(.*?)\)', rec)[0]
digest = re.findall(r'== (.*?)\n', rec)[0]
print("suffix: {suffix} \ndigest: {digest}")
print('Calculating hash...')
for i in product(table, repeat=3):
    prefix = ''.join(i)
    guess = prefix + suffix
    if sha384(guess.encode()).hexdigest() == digest:
        print(guess)
        break
io.sendlineafter(b'Give me XXX:', prefix.encode())
io.recvuntil(',_|\n\n')


p0 = b'Blowfish_w0rld'
c0 = base64.b64decode(io.recvline().strip())
sendIV, c0 = c0[:8], c0[8:]
target = b'get_flag'

iv = []
for idx, val in enumerate(target):
    iv.append(sendIV[idx] ^ target[idx] ^ p0[idx])
iv = bytes(iv)
crafted_message = base64.b64encode(iv+c0)
flag = ''
for boff in range(0, 48, 8):
    for off in range(7, -1, -1):
        io.sendline(crafted_message)
        io.sendline('\x00'*off)
        res = base64.b64decode(io.recvline())
        target = res[boff:boff+8]

for i in range(33, 128):
    io.sendline(crafted_message)
    io.sendline('\x00'*off+flag+chr(i))
    res = base64.b64decode(io.recvline())[boff:boff+8]
    if res == target:
        flag += chr(i)

print(flag)
```