week1

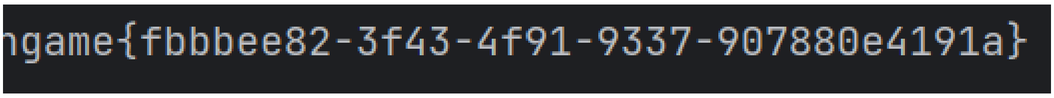
crypto

PRNG

看题，是一道伪随机数生成，观察PRNG代码，后面bit的信息是和前面在mask位置上1的个数有关，如果为奇数，那么后面一位加1，不然加0。所以output的前32位就可以推断原来的信息。

num函数作用判断后面bit是1还是0；如果前31位参与函数就能输出正确结果就说明上一位位是0，不然为1

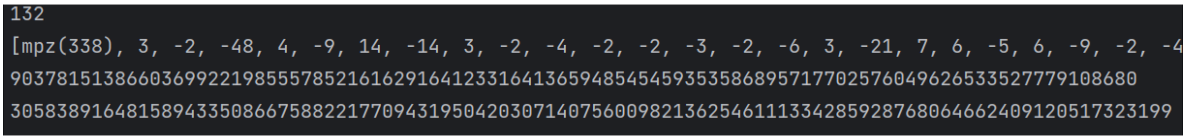
```python
import uuid
mask=0b10001001000010000100010010001001

output=[]  #output太大省略
ans=''
flag=''
def num( R):
    i = R & mask
    nextbit = 0
    while i!=0:
        nextbit^=(i%2)
        i=i//2
    return nextbit
for i in range(4):
    R=output[i][:32]
    for j in range(32):
        if num(int(R[:31], 2))==int(R[-1]):
            R="0"+R
        else:
            R="1"+R
        R=R[:32]

    ans+=hex(int(R[:32],2))[2:]
    print(R)

print(ans)
flag='hgame{'+str(ans[:8])+'-'+str(ans[8:12])+'-'+str(ans[12:16])+'-
'+str(ans[16:20])+'-'+str(ans[20:])+'}'
print(flag)
```



```
hgame{fbbbee82-3f43-4f91-9337-907880e4191a}
```

ezmath

Pell方程，连分数分解找到基本解



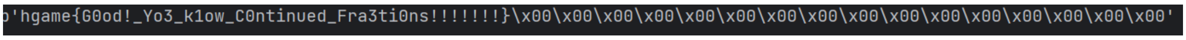
```
132
[mpz(338), 3, -2, -48, 4, -9, 14, -14, 3, -2, -4, -2, -2, -3, -2, -6, 3, -21, 7, 6, -5, 6, -9, -2, -4
9037815138660369922198555785216162916412331641365948545459353586895717702576049626533527779108680
3058389164815894335086675882217709431950420307140756009821362546111334285928768064662409120517323199
```

```
from gmpy2 import *


def Cal_CF(List):
    List.reverse()
    fenmu = 0
    fenzi = 1
    for i in List:
        fenmu, fenzi = fenzi, i * fenzi + fenmu
    return fenmu, fenzi




m = isqrt(D)
x = D ** (0.5)
a = []
a.append(m)
b = m
c = 1
while a[-1] != 2 * a[0]:
    c = (D - b * b) // c
    tmp = (x + b) / c
    a.append(int(tmp))
    b = a[-1] * c - b
print(len(a) - 1)
print(a)
a = a[:-1]
fenmu, fenzi = Cal_CF(a)
print(fenmu)
print(fenzi)
```

找到最小解后理论上来说应该推后面的解，但是觉得才week1，应该解出来最小解就ok了，试了一下出flag了。

```
from Crypto.Util.number import *
from Crypto.Cipher import AES
import random,string,math,gmpy2
enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17g\x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1,U\xfe\xdec\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"

D = 114514
def pad(x):
    return x+b'\x00'*(16-len(x)%16)
def decrypt(KEY):
    cipher= AES.new(KEY,AES.MODE_ECB)
    decrypted =cipher.decrypt(enc)
    return decrypted
from gmpy2 import *

y0=9037815138660369922198555785216162916412331641365948545459353586895717702576049626533527779108680
x0=3058389164815894335086675882217709431950420307140756009821362546111334285928768064662409120517323199
key=pad(long_to_bytes(y0))[:16]
flag=decrypt(key)
```

```
print(flag)
```



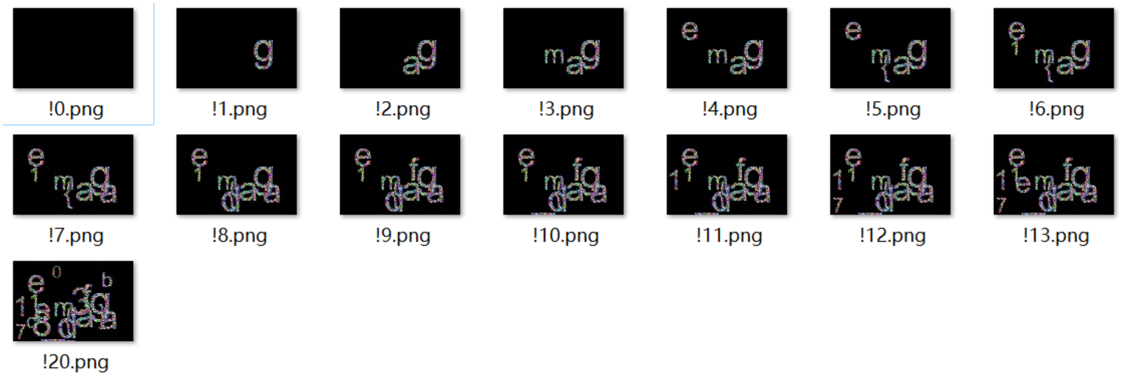
```
b'hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!!}\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

奇怪的图片

把所有图片都异或，那剩下的就是上面的flag的碎片了，再从里面找出第一张图片，第一张图片的上面只有一个h，和其他图片异或之后就会一个字符一个字符的显示出。根据字符出现的顺序就是flag上的顺序。

```python
from PIL import Image, ImageDraw, ImageFont
import threading
import random
import secrets
def xor_images(image1, image2):
    if image1.size != image2.size:
        raise ValueError("Images must have the same dimensions.")
    xor_image = Image.new("RGB", image1.size)
    pixels1 = image1.load()
    pixels2 = image2.load()
    xor_pixels = xor_image.load()
    for x in range(image1.size[0]):
        for y in range(image1.size[1]):
            r1, g1, b1 = pixels1[x, y]
            r2, g2, b2 = pixels2[x, y]
            xor_pixels[x, y] = (r1 ^ r2, g1 ^ g2, b1 ^ b2)
    return xor_image
images={}
for i in range(21):
    images[i]=Image.open("F:\picture\png_out\\{}.png".format(i))
k=0
for i in range(0,21):
    for j in range(0,21):
        image=xor_images(images[i],images[j])
        image.save("F:\picture\png_out\\{}\\{}.png".format(i,j))
```

刚开始还在想为什么看不见hgame的h，后来好好想想才发现确实，第一个字母每张图片都有，整理图片（最后一个字母居然是c，我刚开始在o和d里面抉择）



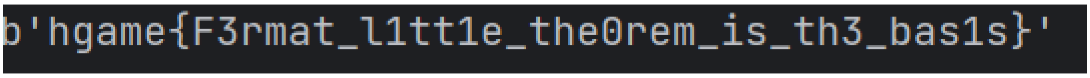
ezRSA

leak1和leak2就是p和q。根据欧拉定理

对任意两个正整数 a, n，若两者素质，则：$a^{\varphi(n)} \equiv 1(\mathrm{mod}\ n)$。

所以,p^(q-1)=1(mod q),p^q=p mod q=p mod n

```
from Crypto.Util.number import *

p=leak1=149127170073611271968182576751290331559018441805725310426095412837589227
670757540743929865853650399839102838431507200744724939659463200158012469676979998
76964190509008427982256658618123311136328924387427242029164160602665815901690638
6768829928898573410412763223217565735269789838344132347745065817972772890866 9
q=leak2=116122992714670915381309916967490436489020001172880644167179915467021794
8929279772720805966417855691191342590375223883351980431522061502591034855745 5881
642474020473621555193348258394195999462535658120105453452939578174433863102142 37
0317114645666343295584359854812259330878224522079201871650853849740257670 9461
c=1052948186753252003425805677386407401702701957804186624540064784023025166165 29
99709715919620810933437191661180003295923273655675729588558899592524235622728816
065501918076120812236580344991140980991532347991252705288633014913479970610056 84
554352359132417756706194892255227523548661551491393212543654399164260702868976 26
936173052467164927831168130703555126069716266455949618505675863403897058213148 42
096465631886812281289843132258131809773797777049358789182212570606252509790830 99
426313202009415364629679352297563219191246391989898834928228497291993276195260 33
79733234575351624039162440021940592552768579639977713099971

e=0x10001
n=p*q
phi=(p-1)*(q-1)
d=pow(e,-1,phi)
m=pow(c,d,n)
ans=long_to_bytes(m)
print(ans)
```

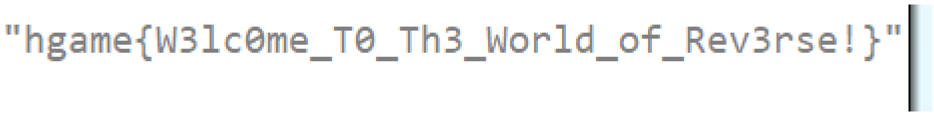


```
b'hgame{F3rmat_l1tt1e_the0rem_is_th3_bas1s}'
```

REVERSE

ezIDA

用IDA打开即得到flag



```
"hgame{W3lc0me_T0_Th3_World_of_Rev3rse!}"
```

PWN

EzSign

nc一下就好了

MISC

SignIn

用ps改一下照片尺寸