小菜鸡刷CTF

原创

大白羊想学习 于 2019-07-19 09:33:44 发布 1405 收藏 1

分类专栏： CTF 文章标签： CTF 菜鸟

本文链接：https://blog.csdn.net/weixin_44728238/article/details/96431085


CTF 专栏收录该内容

4 篇文章 0 订阅

订阅专栏

一个小菜鸡的CTF成长史

本人小菜鸡一枚，刚入门CTF，不对，应该是脚刚刚沾上了CTF的边边，就只是在刷攻防世界中的题目。无奈脑子太笨，记性太差，只好将题目整理下来，代码多有借鉴，求勿喷~~

新手练习

题目一：base64

题目来源： poxlove3
题目描述：元宵节灯谜是一种古老的传统民间观灯猜谜的习俗。 因为谜语能启迪智慧又饶有兴趣，灯谜增添节日气氛，是一项很有趣的活动。 你也很喜欢这个游戏，这不，今年元宵节，心里有个黑客梦的你，约上你青梅竹马的好伙伴小鱼， 来到了cyberpeace的攻防世界猜谜大会，也想着一展身手。 你们一起来到了小孩子叽叽喳喳吵吵闹闹的地方，你俩抬头一看，上面的大红灯笼上写着一些奇奇怪怪的字符串，小鱼正纳闷呢，你神秘一笑，我知道这是什么了。
题目附件：
Y3liZXJwZWFjZXtXZWxjb21lX3RvX25ld19Xb3JsZCF9

解题：
题目中已经提示是使用base64编码了，但是，之后的场景中如何来判断是否使用base64编码呢，通常而言，使用base64编码的文字具有以下特征：

1. 字符串只可能包含A-Z，a-z，0-9，+，/，=字符
2. 字符串长度是4的倍数
3. =只会出现在字符串最后，可能没有或者一个等号或者两个等号

那么回到这道题，对于base64编码的文字我们要如何找到原文字呢，由于网上有许多的base64在线工具，使用工具是很方便的，比如说：https://base64.supfree.net/
可以轻松地得到结果cyberpeace{Welcome_to_new_World!}
或者也可以使用python代码解码：

```
import base64
with open("crypto1.txt") as f:
    source = f.read()
print(base64.b64decode(source))
```

题目二：Caesar

题目来源： poxlove3
题目描述：你成功的解出了来了灯谜，小鱼一脸的意想不到“没想到你懂得这么多啊！”你心里面有点小得意，“那可不是，论学习我没你成绩好轮别的我知道的可不比你少，走我们去看看下一个”你们继续走，看到前面也是热热闹闹的，同样的大红灯笼高高挂起，旁边呢好多人叽叽喳喳说个不停。你一看 大灯笼，上面还是一对字符，你正冥思苦想呢，小鱼神秘一笑，对你说道，我知道这个的答案是什么了
题目附件：oknqdbqmoq{kag_tmhq_xqmdzqp_omqemd_qzodkbfuaz}

解题：
这道题使用的是caesar凯撒密码，这是一种字母的置换密码，看到密文中有一对{}且恰好在文字后方，就可以想到可能是置换密码。
凯撒密码使用在线工具是非常方便的，例如：https://www.qqxiuzi.cn/bianma/kaisamima.php
答案是cyberpeace{you_have_learned_caesar_encryption}
同样的，也可以利用代码来进行解密，解密的关键就是替代每个明文字母的是字母表中位移多少的字母，即C=(P+k) mod 26中的k

```
with open ("crypto2.txt") as f:
    data=f.read()
print(data)
word=[]
for i in range(0,26):
    print(i)
    word=[]
    for j in data:
        if(j>='a' and j<='z') :
            word.append(chr(((ord(j)-97+i)%26)+97))
        elif (j>='A' and j<='Z'):
            word.append(chr(((ord(j)-65+i)%26)+65))
        else:
            word.append(j)
    print(''.join(word))
```

题目三：Morse

题目来源： poxlove3
题目描述：小鱼得意的瞟了你一眼，神神气气的拿走了答对谜语的奖励，你心里暗暗较劲 想着下一个谜题一定要比小鱼更快的解出来。不知不觉你们走到了下一个谜题的地方，这个地方有些奇怪。 上面没什么提示信息，只是刻着一些0和1，感觉有着一些奇怪的规律，你觉得有些熟悉，但是就是想不起来 这些01代表着什么意思。一旁的小鱼看你眉头紧锁的样子，扑哧一笑，对你讲“不好意思我又猜到答案了。”(flag格式为cyberpeace{xxxxxxxxxx},均为小写)
题目附件：
11 111 010 000 0 1010 111 100 0 00 000 000 111 00 10 1 0 010 0 000 1 00 10 110
摩斯密码解密过程实际上就是一对一的过程,用字典存好摩斯密码的东西,主键为摩斯串,值为字符，然后根据摩斯串把相应的字符打印出来。
同样的，附上摩斯密码的在线解密网址：https://www.jb51.net/tools/morse.html
以及解密代码

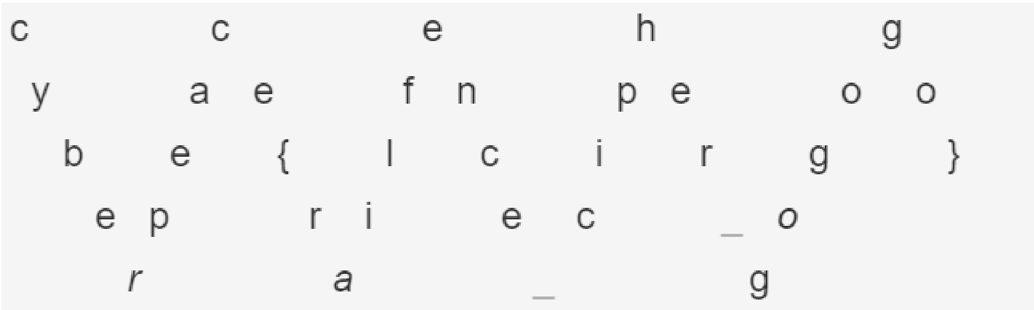
```
from __future__ import print_function
with open("crypto3.txt") as f:
    data=f.read()
s = data.replace('1','-').replace('0','.').split(" ")
print(s)
dict = {'.-': 'A',
        '-...': 'B',
        '-.-.': 'C',
        '-..':'D',
        '.':'E',
        '..-.':'F',
        '--.': 'G',
        '....': 'H',
        '..': 'I',
        '.---':'J',
        '-.-': 'K',
        '.-..': 'L',
        '--': 'M',
        '-.': 'N',
        '---': 'O',
        '.--.': 'P',
        '--.-': 'Q',
        '.-.': 'R',
        '...': 'S',
        '-': 'T',
        '..-': 'U',
        '...-': 'V',
        '.--': 'W',
        '-..-': 'X',
        '-.--': 'Y',
        '--..': 'Z',
        '.----': '1',
        '..---': '2',
        '...--': '3',
        '....-': '4',
        '.....': '5',
        '-....': '6',
        '--...': '7',
        '---..': '8',
        '----.': '9',
        '-----': '0',
        '..--..': '?',
        '-..-.': '/',
        '-.--.-': '()',
        '-....-': '-',
        '.-.-.-': '.',
        '..--.-': '-'
        };
for item in s:
    print (dict[item],end='')
```

最终得到的结果为：MORSECODEISSOINTERESTING，按照题目要求的格式写入就可以啦~

题目四：Railfence

题目来源： poxlove3
题目描述：被小鱼一连将了两军，你心里更加不服气了。两个人一起继续往前走， 一路上杂耍卖艺的很多，但是你俩毫无兴趣，直直的就冲着下一个谜题的地方去了。 到了一看，这个谜面看起来就已经有点像答案了样子了，旁边还画着一张画，是一副农家小院的 图画，上面画着一个农妇在栅栏里面喂5只小鸡，你嘿嘿一笑对着小鱼说这次可是我先找到答案了。
题目附件：ccehgyaefnpeoobe{lcirg}epriec_ora_g

这道题的加密后文件中本身就含有{}，但是又不是在最后，这就说明很有可能是打乱了顺序的，一种相符合的加密方式是栅栏密码。
然而这道题的栅栏密码与常见的栅栏密码并不一样，并非是横着排列的，而是以W型排列，至于为什么…可能大佬懂吧，反正我在做的时候是很难受的。
在题目本身的writeup中有一个在线工具http://ctf.ssleye.com/railfence.html，但是需要付费，土豪自便~~
小菜鸡加小穷B就只好自己做咯，由于总共是35个字符，所以可以猜测key为5或者7。但是这道题我并没有写出代码，这主要是因为为了能让最后一个}在最后一个字符，并不是完全按照W型写的，总之我并不是很理解这种操作，如果有大佬也可以帮我讲



解一下，不胜感激。
结果就是：cyberpeace{railfence_cipher_gogogo}

题目五：不仅仅是Morse

题目来源： poxlove3
题目描述：“这个题目和我们刚刚做的那个好像啊但是为什么按照刚刚的方法做出来答案却不对呢”，你奇怪的问了问小鱼，“可能是因为还有一些奇怪的加密方式在里面吧，我们在仔细观察观察”。两个人 安安静静的坐下来开始思考，很耐心的把自己可以想到的加密方式一种种的过了一遍，十多分钟后两个人 异口同声的说“我想到了！”
题目附件：
–/.-/-.–/…–.-/-…/./…–.-/…/.-/…-/./…–.-/.-/-./—/-/…/./.-./…–.-/-…/./-.-./—/-…/./…/…/…/…/.-/.-/.-/.-/.-/-…/.-/.-/-…/-…/-…/.-/.-/-…/-…/.-/.-/.-/.-/.-/.-/.-/.-/-…/.-/.-/-…/.-/-…/.-/.-/.-/.-/.-/.-/.-/-…/-…/.-/-…/.-/.-/.-/-…/-…/.-/.-/.-/-…/-…/.-/-…/.-/.-/.-/.-/-…/.-/-…/.-/.-/-…/.-/.-/.-/-…/-…/.-/-…/.-/.-/-…/.-/.-/.-/-…/.-/.-/-…/.-/.-/-…/-…/.-/-…/.-/.-/-…/-…/.-/.-/-…/-…/-…/.-/.-/-…/.-/-…/.-/-…/.-/.-/-…/-…/.-/.-/.-/-…/-…/.-/-…/.-/.-/.-/-…/.-/.-/-…/.-/.-/.-/.-/-…/-…/.-/-…/-…/.-/.-/-…/-…/.-/-…/.-/.-/-…/.-/.-/.-/-…/.-/.-/-…/.-/.-/-…/.-/-…/.-/-…/.-/.-/.-/.-/-…/-…/.-/-…/.-/.-/-…/-…/.-

解题：
首先看到这个格式，首先想到的就是莫斯密码，可以得到莫斯密码解密后的结果是：
MAY-BE-HAVE-ANOTHER-DECODEHHHHAAAAABAABBBAABBAAAAAAAAABAABABAAAAAAABBABAAABBAAABBAABAAAABABAABAAABBABAAABAAABAABABBAABBBABAAABABABBAAABBABAAABAABAABAAAABBABBAABBAABAABAAABAABAABAABABAABBABAAAABBABAABBA
很显然后面的A、B组成的部分还需要再次解密,只含有A、B的密码一般为培根密码，在线解码网站为
http://rumkin.com/tools/cipher/baconian.php
代码（在网上找到的，可以加密、解密，比我写得好hhh）

```
import re

alphabet = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

first_cipher = ["aaaaa","aaaab","aaaba","aaabb","aabaa","aabab","aabba","aabbb","abaaa","abaab","ababa","ababb",
```

```
"abbaa","abbab","abbba","abbbb","baaaa","baaab","baaba","baabb","babaa","babab","babba","babbb","bbaaa","bbaab"]

second_cipher = ["aaaaa","aaaab","aaaba","aaabb","aabaa","aabab","aabba","aabbb","abaaa","abaaa","abaab","ababa"
,"ababb","abbaa","abbab","abbba","abbbb","baaaa","baaab","baaba","baabb","baabb","babaa","babab","babba","babbb"
]

def encode():
    upper_flag = False # 用于判断输入是否为大写
    string = input("please input string to encode:\n")
    if string.isupper():
        upper_flag = True
        string = string.lower()
    e_string1 = ""
    e_string2 = ""
    for index in string:
        for i in range(0,26):
            if index == alphabet[i]:
                e_string1 += first_cipher[i]
                e_string2 += second_cipher[i]
                break
    if upper_flag:
        e_string1 = e_string1.upper()
        e_string2 = e_string2.upper()
    print ("first encode method result is:\n"+e_string1)
    print ("second encode method result is:\n"+e_string2)
    return


def decode():
    upper_flag = False  # 用于判断输入是否为大写
    e_string = input("please input string to decode:\n")
    if e_string.isupper():
        upper_flag = True
        e_string = e_string.lower()
    e_array = re.findall(".{5}",e_string)
    d_string1 = ""
    d_string2 = ""
    for index in e_array:
        for i in range(0,26):
            if index == first_cipher[i]:
                d_string1 += alphabet[i]
            if index == second_cipher[i]:
                d_string2 += alphabet[i]
    if upper_flag:
        d_string1 = d_string1.upper()
        d_string2 = d_string2.upper()
    print ("first decode method result is:\n"+d_string1)
    print ("second decode method result is:\n"+d_string2)
    return


if __name__ == '__main__':
    print ("\t\tcoding by qux")
    while True:
        print ("\t*******Bacon Encode_Decode System*******")
        print ("input should be only lowercase or uppercase,cipher just include a,b(or A,B)")
        print ("1.encode\n2.decode\n3.exit")
        s_number = input("please input number to choose\n")
        if s_number == "1":
            encode()
```

```
            input()
        elif s_number == "2":
            decode()
            input()
        elif s_number == "3":
            break
        else:
            continue
```

得到的结果为：ATTACKANDDEFENCEWORLDISINTERESTING，把两次解密的结果组合起来就好啦~

题目六：easy_RSA

题目来源： poxlove3
题目描述：解答出来了上一个题目的你现在可是春风得意，你们走向了下一个题目所处的地方 你一看这个题目傻眼了，这明明是一个数学题啊！！！可是你的数学并不好。扭头看向小鱼，小鱼哈哈一笑 ，让你在学校里面不好好听讲现在傻眼了吧~来我来！三下五除二，小鱼便把这个题目轻轻松松的搞定了
题目附件：
在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17
求解出d

解题：
是令人头大的RSA啊啊啊，不过这仿佛是最简单的一道RSA题目了，附代码：

```
import gmpy2
p=gmpy2.mpz(473398607161)
q =gmpy2.mpz(4511491)
e = 17
mod=gmpy2.mpz((p-1)*(q-1))
d=gmpy2.invert(e,mod)
print(d)
```

题目七：混合编码

题目来源： poxlove3
题目描述：经过了前面那么多题目的历练，耐心细致在解题当中是 必不可少的品质，刚巧你们都有，你和小鱼越来越入迷。那么走向了下一个题目，这个题目好长 好长，你知道你们只要细心细致，答案总会被你们做出来的，你们开始慢慢的尝试，慢慢的猜想 ，功夫不负有心人，在你们耐心的一步步的解答下，答案跃然纸上，你俩默契一笑，相视击掌 走向了下面的挑战。
题目附件：
JiM3NjsmlzEyMjsmlzY5OyYjMTIwOyYjNzk7JiM4MzsmlzU2OyYjMTIwOyYjNzc7JiM2ODsmlzY5OyYjMTE4OyYjNzc7JiM4NDsmlzY1OyYjNTI7JiM3NjsmlzEyMjsmlzEwNzsmlzUzOyYjNzY7JiMxMjI7JiM2OTsmlzEyMDsmlzc3OyYjODM7JiM1NjsmlzEyMDsmlzc3OyYjNjg7JiMxMDc7JiMxMTg7JiM3Nzsmlzg0OyYjNjU7JiMxMjA7JiM3NjsmlzEyMjsmlzY5OyYjMTIwOyYjNzg7JiMxMDU7JiM1NjsmlzEyMDsmlzc3OyYjODQ7JiM2OTsmlzExODsmlzc5OyYjODQ7JiM5OTsmlzExODsmlzc3OyYjODQ7JiM2OTsmlzUwOyYjNzY7JiMxMjI7JiM2OTsmlzEyMDsmlzc4OyYjMTA1OyYjNTY7JiM1Mzsmlzc4OyYjMTIxOyYjNTY7JiM1Mzsmlzc5OyYjODM7JiM1NjsmlzEyMDsmlzc3OyYjNjg7JiM5OTsmlzExODsmlzc5OyYjODQ7JiM5OTsmlzExODsmlzc3OyYjODQ7JiM2OTsmlzExOTsmlzc2OyYjMTIyOyYjNjk7JiMxMTk7JiM3NzsmlzY3OyYjNTY7JiMxMjA7JiM3NzsmlzY4OyYjNjU7JiMxMjg7JiM3NzsmlzY4OyYjNjU7JiM2NTsmlzY5OyYjMTE5OyYjNzc7JiMxMDU7JiM1NjsmlzEyMDsmlzc3OyYjNjg7JiM2OTsmlzExODsmlzc3OyYjODQ7JiM2OTsmlzExOTsmlzc2OyYjMTIyOyYjNjA3OyYjNTM7JiM3NjsmlzEyMjsmlzY5OyYjMTE5OyYjNzc7JiM4MzsmlzU2OyYjMTIwOyYjNzc7JiM4NDsmlzEwNzsmlzExODsmlzc3OyYjODQ7JiM2OTsmlzEyMDsmlzc2OyYjMTIyOyYjNjk7JiMxMjA7JiM3ODsmlzY3OyYjNTY7JiMxMjA7JiM3NzsmlzY4OyYjMTAzOyYjMTE4OyYjNzc7JiM4NDsmlzY1OyYjMTE5Ow==

解题：有等号！！！必然是首先用base64解码了，得到结果：
LzExOS8xMDEvMTA4Lzk5LzExMS8xMDkvMTAxLzExNi8xMTEvOTcvMTE2LzExNi85Ny85OS8xMDcvOTcvMTEwLzEwMC8xMDAvMTAxLzEwMi8xMDEvMTEwLzk5LzEwMS8xMTkvMTExLzExNC8xMDgvMTAw
用Unicode解码，得到：
LzExOS8xMDEvMTA4Lzk5LzExMS8xMDkvMTAxLzExNi8xMTEvOTcvMTE2LzExNi85Ny85OS8xMDcvOTcvMTEwLzEwMC8xMDAvMTAxLzEwMi8xMDEvMTEwLzk5LzEwMS8xMTkvMTExLzExNC8xMDgvMTAw
再次用base64解码，得到：
/119/101/108/99/111/109/101/116/111/97/116/116/97/99/107/97/110/100/100/101/102/101/110/99/101/119/111/114/108/100
利用ASCII进行转换，得到字符串：
welcometoattackanddefenceworld
Unicode的在线解码网站：
http://tool.chinaz.com/tools/unicode.aspx
ASCII在线转化网址：
http://ctf.ssleye.com/jinzhi.html

题目八：Normal_RSA

题目来源： PCTF
题目描述：你和小鱼走啊走走啊走，走到下一个题目一看你又一愣，怎么还是一个数学题啊 小鱼又一笑，hhhh数学在密码学里面很重要的！现在知道吃亏了吧！你哼一声不服气，我知道数学 很重要了！但是工具也很重要的，你看我拿工具把他解出来！你打开电脑折腾了一会还真的把答案 做了出来，小鱼有些吃惊，向你投过来一个赞叹的目光
题目附件：
我们将压缩包解压发现，这道题给了两个文件，一个是加密过的的flag.enc，另一个是公钥pubkey.pem。
flag.enc中的内容为：m>愤#铖訃竞x爣?溯?lm越扐? y
public.pem中的内容为：
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDr
yigb/+l/vjDdAgMBAAE=
-----END PUBLIC KEY-----

解题：
首先我们需要找到私钥，这里给出了公钥，首先我们使用openssl提取出pubkey.pem中的参数：
openssl rsa -pubin -text -modulus -in warmup -in pubkey.pem
得到了公钥内容如下：

```
RSA Public-Key: (256 bit)
Modulus:
    00:c2:63:6a:e5:c3:d8:e4:3f:fb:97:ab:09:02:8f:
    1a:ac:6c:0b:f6:cd:3d:70:eb:ca:28:1b:ff:e9:7f:
    be:30:dd
Exponent: 65537 (0x10001)
Modulus=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDr
yigb/+1/vjDdAgMBAAE=
-----END PUBLIC KEY-----
```


Modulus中为两个大素数P，Q的乘积，即
N=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
对于这种N不是很大的情况，我们可以直接分解得到P，Q，可以使用在线工具http://www.factordb.com/index.php
可以得到P、Q分别为275127860351348928173285174381581152299，319576316814478949870590164193048041239
由P，Q我们可以得到私钥文件，可以使用RSAtool来获取private.pem，指令为：
python rsatool.py -o private.pem -e 65537 -p 275127860351348928173285174381581152299 -q 319576316814478949870590164193048041239
用private.pem就可以解出明文了：
openssl rsautl -decrypt -in flag.enc -inkey private.pem
最后结果为：PCTF{256b_i5_m3dium}

题目九：轮转机加密

题目来源： ISCC2017
题目描述：你俩继续往前走，来到了前面的下一个关卡，这个铺面墙上写了好多奇奇怪怪的 英文字母，排列的的整整齐齐，店面前面还有一个大大的类似于土耳其旋转烤肉的架子，上面一圈圈的 也刻着很多英文字母，你是一个小历史迷，对于二战时候的历史刚好特别熟悉，一拍大腿：“嗨呀！我知道 是什么东西了！”
题目附件：
1: < ZWAXJGDLUBVIQHKYPNTCRMOSFE <
2: < KPBELNACZDTRXMJQOYHGVSFUWI <
3: < BDMAIZVRNSJUWFHTEQGYXPLOCK <
4: < RPLNDVHGFCUKTEBSXQYIZMJWAO <
5: < IHFRLABEUOTSGJVDKCPMNZQWXY <
6: < AMKGHIWPNYCJBFZDRUSLOQXVET <
7: < GWTHSPYBXIZULVKMRAFDCEONJQ <
8: < NOZUTWDCVRJLXKISEFAPMYGHBQ <
9: < XPLTDSRFHENYVUBMCQWAOIKZGJ <
10: < UDNAJFBOWTGVRSCZQKELMXYIHP <
11： < MNBVCXZQWERTPOIUYALSKDJFHG <
12： < LVNCMXZPQOWEIURYTASBKJDFHG <
13： < JZQAWSXCDERFVBGTYHNUMKILOP <

密钥为：2,3,7,5,13,12,9,1,8,10,4,11,6
密文为：NFQKSEVOQOFNP

解题：
此处的轮转法为托马斯.杰斐逊轮转法。
首先，密钥与1-13组文字有关，可以根据密钥调整每组文字的顺序，例如：第2组文字在第1行；
接下来，根据密文调整每行字符串，第一个密文为N，则表示在第二组文字中，将N及后面的字母提到最前面，前方的文字放在后面；
以此类推…

```
import re
init = '1: < ZWAXJGDLUBVIQHKYPNTCRMOSFE < 2: < KPBELNACZDTRXMJQOYHGVSFUWI < 3: < BDMAIZVRNSJUWFHTEQGYXPLOCK < 4:
 < RPLNDVHGFCUKTEBSXQYIZMJWAO < 5: < IHFRLABEUOTSGJVDKCPMNZQWXY < 6: < AMKGHIWPNYCJBFZDRUSLOQXVET < 7: < GWTHSPY
BXIZULVKMRAFDCEONJQ < 8: < NOZUTWDCVRJLXKISEFAPMYGHBQ < 9: < XPLTDSRFHENYVUBMCQWAOIKZGJ < 10: < UDNAJFBOWTGVRSCZ
QKELMXYIHP < 11 < MNBVCXZQWERTPOIUYALSKDJFHG < 12 < LVNCMXZPQOWEIURYTASBKJDFHG < 13 < JZQAWSXCDERFVBGTYHNUMKILOP
 <'
cipher_text = 'NFQKSEVOQOFNP'
# 将sss转化为列表形式
content=re.findall(r'< (.*?) <',init,re.S)
# re.S:DOTALL，此模式下，"."的匹配不受限制，可匹配任何字符，包括换行符#.*?可以为任意字符
key=[2,3,7,5,13,12,9,1,8,10,4,11,6]
final=[]
for i in range(0,13):
    result=''
    text=content[key[i]-1]
    index=text.index(cipher_text[i])
    for j in range(index,26):
        result+=text[j]
    for j in range(0,index):
        result+=text[j]
    final.append(result)
print(final)
for i in range(0,26):
    for j in range(0,13):
        print(final[j][i],end='')
    print('\n')
```

题目十：easychallenge

题目来源： NJUPT_CTF
题目描述：你们走到了一个冷冷清清的谜题前面，小鱼看着题目给的信息束手无策，丈二和尚摸不着头脑，你嘿嘿一笑，拿出来了你随身带着的笔记本电脑，噼里啪啦的敲起来了键盘，清晰的函数逻辑和流程出现在 了电脑屏幕上，你敲敲键盘，更改了几处地方，运行以后答案变出现在了电脑屏幕上。
题目附件：

```
import base64
def encode1(ans):
    s = ''
    for i in ans:
        x = ord(i) ^ 36
        x = x + 25
        s += chr(x)
    return s
def encode2(ans):
    s = ''
    for i in ans:
        x = ord(i) + 36
        x = x ^ 36
        s += chr(x)
    return s
def encode3(ans):
    return base64.b32encode(ans)
flag = ' '
print 'Please Input your flag:'
flag = raw_input()
final = 'UC7KOWVXWVNKNIC2XCXKHKK2W5NLBKNOUOSK3LNNVWW3E==='
if encode3(encode2(encode1(flag))) == final:
    print 'correct'
else:
    print 'wrong'
```

解题：

这道题给出的显然是一个加密的代码，final是加密得到的密文，而flag是明文。为了获取flag，我们必然是要根据加密算法写出解密的算法，也就是他的逆算法。解密算法如下：

```
import base64
def decode1(ans):
    s = ''
    for i in ans:
        x=ord(i)-25
        x=x^36
        s +=chr(x)
    return s
def decode2(ans):
    s = ''
    for i in ans:
        print(i)
        x=ord(i)^36
        x=x-36
        s += chr(x)
    return s
def decode3(ans):
    return base64.b32decode(ans)
final = 'UC7KOWVXWVNKNIC2XCXKHKK2W5NLBKNOUOSK3LNNVWW3E==='
print(decode1(decode2(decode3(final))))
```

题目十一：幂数加密

题目来源： CFF2016

题目描述：你和小鱼终于走到了最后的一个谜题所在的地方，上面写着一段话“亲爱的朋友， 很开心你对网络安全有这么大的兴趣，希望你一直坚持下去，不要放弃 ，学到一些知识， 走进广阔的安全大世界”，你和小鱼接过谜题，开始了耐心细致的解答。

题目附件：

884210122048022440401422420248012 2

解答：

既然加密后的文字是2的幂的和，这个题中又是以0为界分割的，那么代码如下咯：

```
a="884210122048022440401422420248012 2"
a=a.split("0")
flag=''
for i in range(0,len(a)):
     str = a[i]
     list=[]
     sum=0
     for j in str:
        list.append(j)
        length = len(list)
     for k in range(0,length):
        sum+=int(list[k])
     flag+=chr(sum+64)
print(flag)
```

题目十二：easy_ECC

题目来源： XUSTCTF2016

题目描述：转眼两个人又走到了下一个谜题的地方，这又是一种经典的密码学加密方式 而你刚好没有这个的工具，你对小鱼说“小鱼我知道数学真的很重要了，有了工具只是方便我们使用 懂了原理才能做到，小鱼你教我一下这个缇努怎么做吧！”在小鱼的一步步带领下，你终于明白了ECC 的基本原理，成功的解开了这个题目，两个人相视一笑，快步走向了下一个题目所在的位置。

题目附件：

已知椭圆曲线加密Ep(a,b)参数为

p = 15424654874903 a = 16546484 b = 4548674875

G(6478678675,5636379357093)

私钥为 k = 546768

求公钥K(x,y)

解题：

椭圆曲线的题目真的是无缘，有一种难受叫做不会就是不会，看多少遍都还是不会…有机会再做吧…