mixed cipher

这道题的加密方法是先用一个字母替换密码得到密文1，之后将密文1通过一个固定的key异或得到flag。所以这道题也是一道通过异或的方法加密的题目，所以只要知道密钥就可以得到明文。

那么怎么得到key呢？我们发现这道题的源代码有一个很重要的内容，就是key的长度小于50，并且均由字母、标点、空格和换行符构成，这些字母的acsii码都是有一定范围的。所以，我们可以使用枚举的方法，将每一个byte的可能的值都列举出来。

由于key的长度小于50，我们可以使用枚举的方法，根据上面得到的每一位的可能的值，把每一种key长度下的可能性列举出来。最后得到了唯一的key的长度的确定值：29。

```
import random
from string import *
from itertools import cycle

all_str = digits + ascii_letters + punctuation + ' \n'

f = open("output", 'rb')
a = f.read()

ans_list = []
min_len = 100
length = len(a)

for c in a:
    cur_ans = []
    for i in all_str:
        if chr(c ^ ord(i)) in all_str:
            cur_ans.append(i)
    ans_list.append(cur_ans)
    min_len = min(min_len, len(ans_list))

print('分析完成')

key_list = []
key_flag = []

for cipher_len in range(1, 50):
    key = []
    flag = True
    for shift in range(cipher_len):
        char_list = []
        for c in ans_list[shift]:
            for step in range(shift, length, cipher_len):
                if c not in ans_list[step]:
                    break
            else:
                char_list.append(c)
        key.append(char_list)
```

```
            if len(char_list) == 0:
                flag = False
        key_list.append(key)
        key_flag.append(flag)

    for i in key_list[28]:
        print(i)
```

之后把key的每一位的可能的值通过python脚本得到，发现有一些字母是固定的，通过剩下的字母，我们可以猜测得到这个key的字符串的值。

```
['s']
['a', 'd', 'e', 'f', 'g', 'k', 'n', 'y', 'z']
['c']
['f', 'm', 'p', 'q', 'r', 's', 'v', 'y', '|']
['e', '!']
['t']
['_']
['x']
['1', '2', '3', '4', '5', '6', '8', '9', 'a', 'd', 'f', 'k', 'l', 'm', 'n', 'o',
'p', 's', '!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.',
'/', ':', ';', '=', '>', '?', ' ']
['a', 'm', 'n', 'p', 'q', 'r', 's', 'v', 'y', '{', '|']
['_', '\n']
['k']
['0', '1', '5', '8', '9', 'a', 'd', 'e', 'f', 'g', 'k', 'l', 'n', 'y', 'z', '!',
'"', '#', '$', '%', '&', "'", '(', ')', '*', ',', '-', '/', ';', '<', '=', '>',
'?', '{', ' ']
['y']
['_']
['f']
['1', '2', '3', '4', '5', '6', '7', 'o', '!', '"', '#', '$', '%', '&', "'", '(',
')', '*', '+', ',', '-', '.', '/', ':', ';', '=', '?', ' ']
['r']
['_']
['u']
['_']
['t']
['a', 'd', 'k', 'l', 'm', 'n', 'o', 'p', 's']
['C', 'I', 'Q', 'T', 'V', '@', '[', '\\', ']', '^', '_']
['c']
['m', 'n', 'p', 'q', 'r', 's', 'v', 'y', '{', '|']
['a']
['a', 'b', 'c', 'g', 'h', 'j', 'm', "'", '`', '|']
['b', 'e', 'h', 'i', 'j', 'k', 'o', 't', 'w', '`']
```

通过以上内容，可以得到key为：`secret_xor_key_for_u_to_crack`，之后通过如下的python脚本得到原来的密文：

```
key = 'secret_xor_key_for_u_to_crack'
print(key)

msg = bytes(x ^ ord(y)
            for x, y in zip(a, cycle(key)))
msg = str(msg, encoding='ascii')
print(msg)
```

把输出的密文丢进dcode.fr进行密码替换破解，得到明文和flag。