Unknown date | Thai Nguyen

L.E.I CTF

I used this program to encrypt a flag.
The output
was:n1s4_t1An(f1ctdb@mpl_h3)m3lp3y___Eas

```
print (lambda j,m:(lambda f,t:t if len(t) <= 1
else j([f(f,x)for x in m(j,m(reversed,(lambda
s:zip(*[iter(s)]*(len(s)/2)))(t+"\x01"*
(len(t)%2))))]))(lambda f,t:t if len(t) <= 1 else
j([f(f,x)for x in m(j,m(reversed,(lambda s:
zip(*[iter(s)]*(len(s)/2)))(t+"\x01"*
(len(t)%2))))]),raw_input("Plaintext:")))
(''.join,map).replace("\x01","")
```

I realized it's a obfuscation code so it too hard to reverse code to readable. So i think a ez way to solve this challenge. I think this program will change pos of char so i will find

the pos of each char before encypt.
First i check length of cipher:
len("
n1s4_t1An(f1ctdb@mpl_h3)m3lp3y___Eas")=35

Then i encrypt this type of flag :

ctf(ABCDEFGHIJKLMNOPQRSTUVWXYZ0123)

after i encrypted this type of flag. i would found the pos of each char in this type of flag in encrypted flag. Here is my decrypt code :

```
def enc(plaintext): return (lambda
j,m:(lambda f,t:t if len(t) <= 1 else
j([f(f,x)for x in m(j,m(reversed,
(lambda s:zip(*[iter(s)]*(len(s)/2)))
(t+"\x01"*(len(t)%2))))]))(lambda
f,t:t if len(t) <= 1 else j([f(f,x)for x in
m(j,m(reversed,(lambda s: zip(*
[iter(s)]*(len(s)/2)))(t+"\x01"*
(len(t)%2))))]),plaintext))
(''.join,map).replace("\x01","") def
dcode(cipher):
```

```
flag_type="ctf(ABCDEFGHIJKLMNOPQRSTUVWXYZ0
flag="" for x in flag_type:
flag+=cipher[enc(flag_type).find(x)]
return flag
cip='n1s4_t1An(f1ctdb@mpl_h3)m3lp3y___Eas'
print "\nFLAG : ", dcode(cip)
```

Input :
n1s4_t1An(f1ctdb@mpl_h3)m3lp3y___Eas

FLAG :
ctf(1@mbd4_1nsAn1ty_pl3asE_h3lp_m3)
