

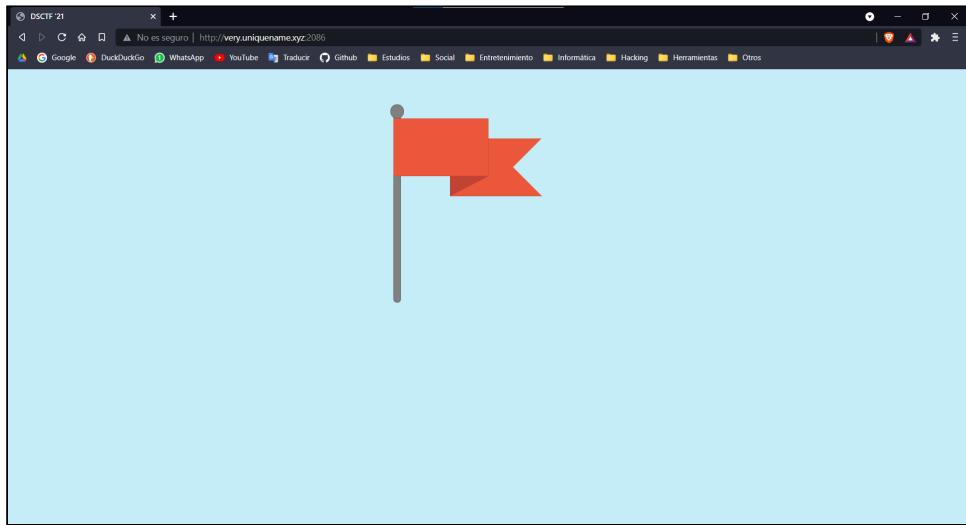
DeconstruCT.F (1-2/10/2021)

Write up escrito por @informaticapau.

Web

Here's a Flag

A quick teaser to get yourself ready for the challenges to come! Just look for/at the flag and perhaps try your hand at some frontend tomfoolery?



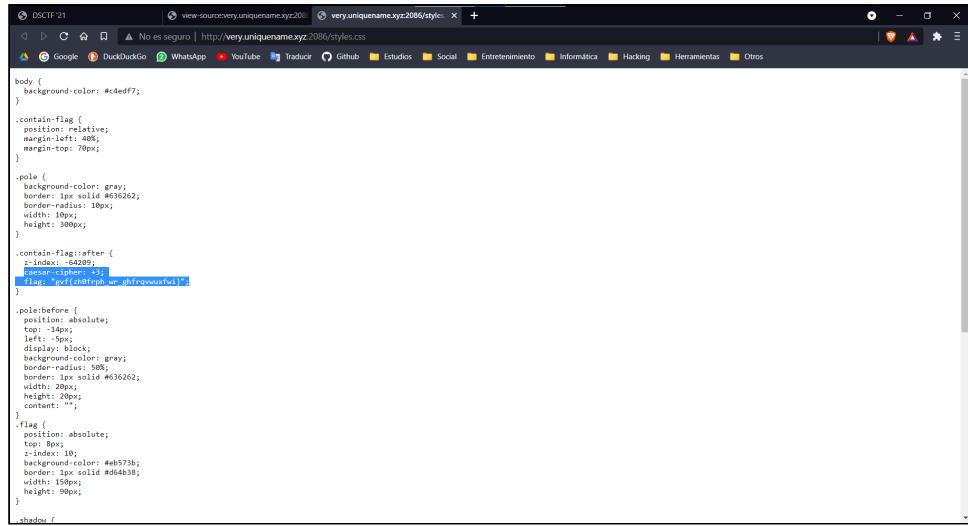
Primero analizo el html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link rel="stylesheet" href="styles.css">
8     <script src="index.js"></script>
9   </head>
10  <body>
11    <div class="contain-flag">
12      <div class="flag"></div>
13      <div class="flag"></div>
14      <div class="shadow"></div>
15      <div class="flag flag-2"></div>
16    </div>
17    <!-- dsc{th15_15_n0t_th3_f14g} -->
18  </body>
19 </html>
```

The screenshot shows the browser's developer tools with the "View Source" tab selected. The page content is displayed as a series of numbered lines of HTML code. The code defines a single-element document type declaration, a head section containing meta tags for character encoding, compatibility, and a viewport, and a body section containing a single div element with the class "contain-flag". Inside this div, there are three more div elements: one with class "flag", one with class "flag", and one with class "shadow". A final div element with class "flag flag-2" is also present. A comment block is included at the end of the body section, starting with "dsc{th15_15_n0t_th3_f14g}" followed by a closing comment tag. The browser interface includes a toolbar with various icons and a navigation bar at the top.

La flag `dsc{th15_15_n0t_th3_f14g}` es falsa.

Tras el html analizo el css:



```
body { background-color: #c4edf7; }

.contain-flag {
    position: relative;
    margin-left: 40px;
    margin-top: 70px;
}

.flag {
    background-color: gray;
    border: 1px solid #636262;
    border-radius: 10px;
    width: 10px;
    height: 300px;
}

.contain-flag::after {
    z-index: -64209;
    caesar-cipher: +3;
    flag: "gvf{zh0frph_wr_ghfrqvwuxfwi}";
}

.pole {
    position: absolute;
    top: -10px;
    left: 50px;
    display: block;
    background-color: gray;
    border-radius: 50%;
    border: 1px solid #636262;
    width: 20px;
    height: 20px;
    content: "";
}

.flag {
    position: absolute;
    top: 80px;
    z-index: 10;
    background-color: #eb573b;
    border: 1px solid #d64b38;
    width: 150px;
    height: 90px;
}

.shadow {
}
```

```
.contain-flag::after {
    z-index: -64209;
    caesar-cipher: +3;
    flag: "gvf{zh0frph_wr_ghfrqvwuxfwi}";
}
```

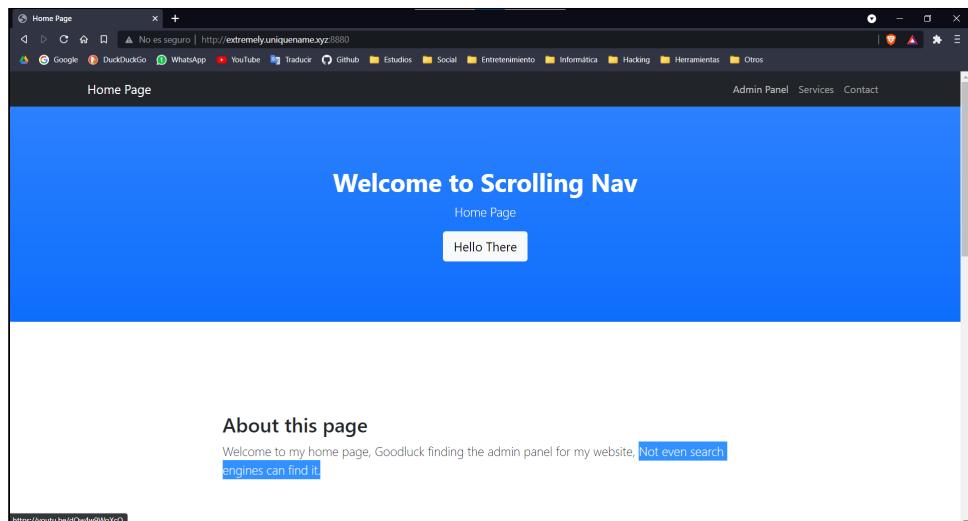
Descifrando el César con clave 3 resulta en `dsc{we0come_to_deconstructf}`.

Sustituyendo el '0' por una 'o' y decifrándola resulta en:

```
dsc{welcome_to_deconstructf}
```

Never gonna lie to you

Trust me, take everything in the home page for face value. I would never lie to you.

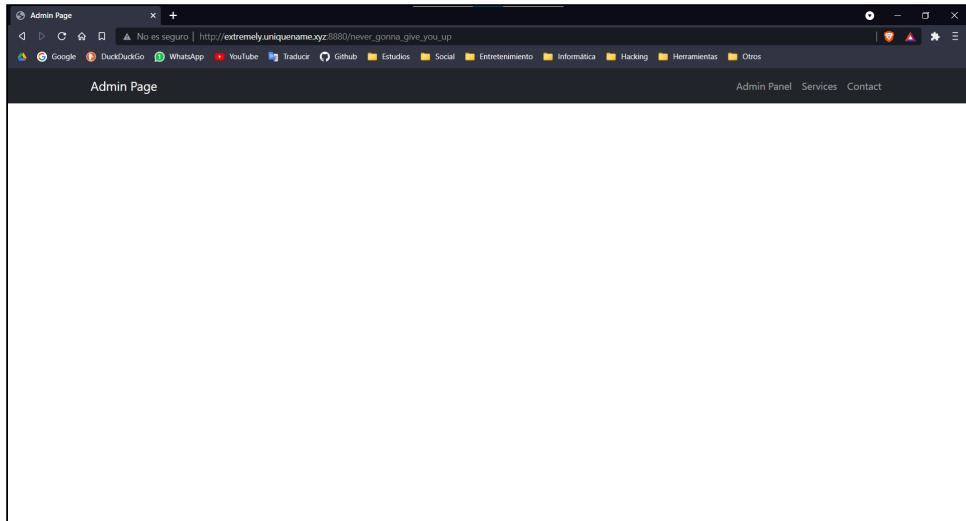


Si alguien intenta acceder al Panel de Admin desde el enlace de la barra de navegación será duramente rickrolleado.

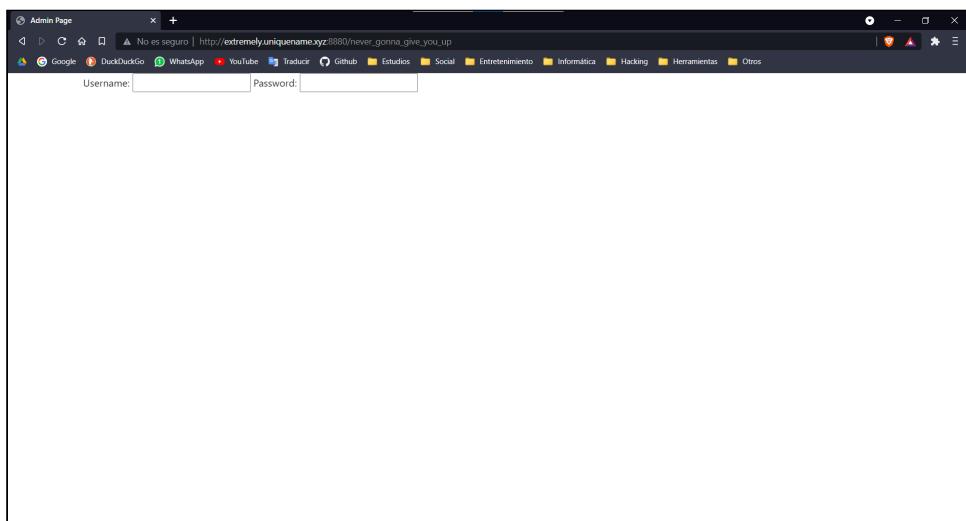
Prestando atención a "Not even search engines can find it." busqué el `robots.txt`.

```
User-agent:*
Disallow:/static/
Disallow:/never_gonna_give_you_up/
```

Después accedía a `/never_gonna_give_you_up` y llegué a la página del Admin.



Tras analizar el html descubrí que detrás de la barra de navegación había un formulario, así que la eliminé.



Le añadí un botón para enviar la solicitud añadiendo `<input type="submit">` y probé a realizar SQL Injection. Con la inyección `' or ''='` conseguí la flag:

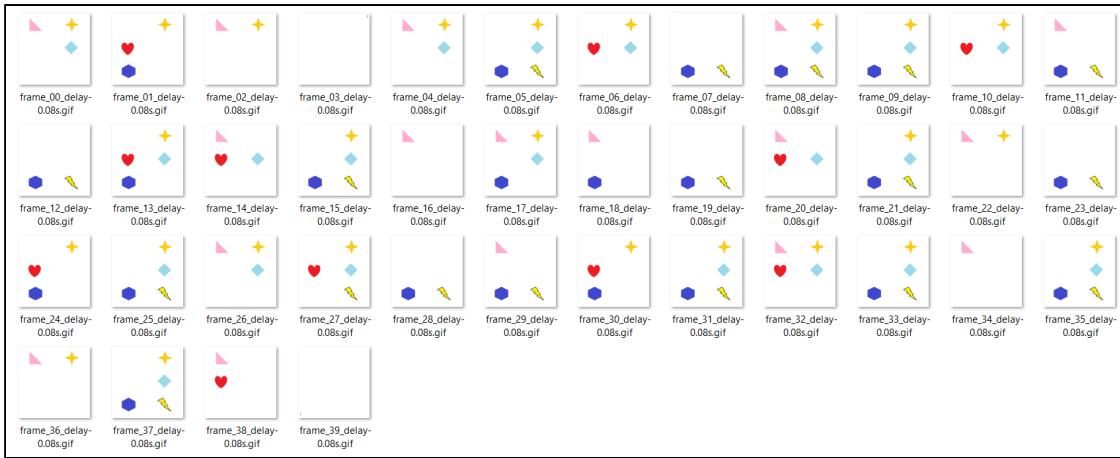
```
dsc{7H15_15_93771N9_0Ld}
```

Cryptography

Stars and Shapes

This might be a difficult question, but I'm sure you can do it with your eyes closed.

Frames de `stars_and_shapes.gif`:



Flag tras decodificarlo de Braille:

```
dsc{d0-y0u-th1nk-h3-s4w-us7132}
```

The Conspiracy

There was once a sailor who traveled to many countries. He was a quirky old man. He said many many things, and most of what he said never made sense to anyone. He considered himself ahead of his time, and said that the people of his time were unworthy of his wisdom. Soon he was lost to the ages, but his diary wasn't. Are you worthy of decoding his wisdom?

Contenido de `diary.txt`:

```
ZHNjeygtMTguMDU1NzI3MjkyODI3NTYSIDE3OC40NTcwMDE0MzEZMTY3NCksKDE5LjAyODI4Mjg1NzUwNTM5MiwgMT
AZLjE0NDI2MDCxMja3MTA3KSWONDIuNTM2NzA1OTkxMjY2MTQ2LCAXLjQ5MzAZNDQ2MTIyNzY5MzMpLCgzOC41ODkz
Njk3MjE3MzU0LCA2OC44MTYzMjuyMzA1ODk2NyksXywONTAUoDUXNTE4OTQ4MjA2Nzk1LCA0LjM2MDE4MDg1MzU4MT
k4NiksKDcunjcxODYzNTM4NDUzMzg2LCazni44MzcynjA5NTY5ODk1MiksXywomzguNjE5NTA2NzQwNTg3MDM1LCAZ
NC44NTUXmjY0MjcxMTAwNCksKDQ3LjQyODA2MTU3MTIzNTQ1LCAXOC450Tg0MjExMDE5MDM2MDIpLCgzMC4xOTM4NZ
E1ODM1MzkyMSwgMzEuMTI0OTY2MTE2MTI1NzMpLF8sKC0wLjIZMtc3NTU5NDI2MTEXNTU4LCAtNzguNTAzMzk2MzA4
Mzk00skskC0xmj44NTQ0NjkzNjczMzQ1NSwgmtMyLjC5MjYyNDMz0DgzMjk4KSw0NDQuNDI0MjmxMjuyNTC3NDM1LC
AyNC4zNTAYNDEXNjYyNzkyMjYpLCgyNC44MTk1NjQ3NDEzOTIzLCaxMjAuOTcyMzY3NTQwMDUwNTgpLCgxOC41ODQ0
NTY5NzQ0MDM0ODcsIC03Mi4zMTgxMjE4OTYxNDC3Mi19
```

Contenido del txt tras descodificarlo desde base64:

```
dsc{(-18.05572729282756, 178.45700143131674), (19.028282857505392, 103.14426071207107),
(42.536705991266146, 1.4930344612276933), (38.5893697217354, 68.81632523058967), _,_
(50.851518948206795, 4.360180853581986), (7.671863538453386, 36.83726095698952), _,_
(38.619506740587035, 34.85512642711004), (47.42806157123545, 18.998421101903602),
(30.19387158353921, 31.12496611612573), _, (-0.23177559426111558, -78.5033963083949),
(-12.85446936733455, 132.79262433883298), (44.424231252577435, 24.350241166279226),
(24.8195647413923, 120.97236754005058), (18.584456974403487, -72.31812189614772)}
```

Coordenadas:

- (-18.05572729282756, 178.45700143131674): Fiyi
- (19.028282857505392, 103.14426071207107): Laos
- (42.536705991266146, 1.4930344612276933): Andorra
- (38.5893697217354, 68.81632523058967): Tayikistan

- _
- (50.851518948206795, 4.360180853581986): Belgic
- (7.671863538453386, 36.83726095698952): Ethiopia
- _
- (38.619506740587035, 34.85512642711004): Turkey
- (47.42806157123545, 18.998421101903602): Hungary
- (30.19387158353921, 31.12496611612573): Egypt
- _
- (-0.23177559426111558, -78.5033963083949): Ecuador
- (-12.85446936733455, 132.79262433883298): Australia
- (44.424231252577435, 24.350241166279226): Romania
- (24.8195647413923, 120.97236754005058): Taiwan
- (18.584456974403487, -72.31812189614772): Haiti

Flag:

```
dsc{FLAT_BE_THE_EARTH}
```

Code Decode

Around 5 years ago, I made this killer program that encodes the string into a ciphertext. The unique feature of this program is that for the same exact plaintext, it generates a different ciphertext every time you run the program. Yesterday I was nosing around in some old stuff and found an encrypted message!

```
2nj1gkma2bv1i0v}221v19vuo19va2bv12{-5x
```

Sadly I realized that I lost the decryption program. I have the encryption program though. Do you think you can help me out and decrypt this message for me?

Tras un estudio exaustivo de `encrypter.py`:

```
from random import choice

inputstring = input("Enter plaintext: ")

def read_encryption_details():
    with open("cypher.txt") as file:
        encrypt_text = eval(file.read())
    encrypt_key = choice(list(encrypt_text.keys()))
    character_key = encrypt_text[encrypt_key]
    return encrypt_key, character_key

def create_encryption(character_key):
    charstring = "abcdefghijklmnopqrstuvwxyz1234567890 _+{}-,.:"
    final_encryption = {}
    for i, j in zip(charstring, character_key):
        final_encryption[i] = j
    return final_encryption

def convert_plaintext_to_cypher(inputstring, final_encryption, encrypt_key):
```

```

cypher_text = ""
for i in inputstring:
    cypher_text += final_encryption[i]
cypher_text = encrypt_key[:3] + cypher_text + encrypt_key[3:]
return cypher_text

encrypt_key, character_key = read_encryption_details()
final_encryption = create_encryption(character_key)
cypher_text = convert_plaintext_to_cypher(
    inputstring, final_encryption, encrypt_key)

print(cypher_text)

```

Decidí crear mi propio descifrador:

```

inputstring = input("Enter plaintext: ")

def read_encryption_details(key):
    with open("cypher.txt") as file:
        encrypt_text = eval(file.read())
        character_key = encrypt_text[key]
    return character_key

def create_decryption(character_key):
    charstring = "abcdefghijklmnopqrstuvwxyz1234567890 _+{}-,.:"
    final_decryption = {}
    for i, j in zip(charstring, character_key):
        final_decryption[j] = i
    return final_decryption

def convert_cypher_to_plaintext(encrypted_text, final_decryption):
    plain_text = ""
    for i in encrypted_text:
        plain_text += final_decryption[i]
    return plain_text

key = inputstring[:3] + inputstring[-3:]
encrypted_text = inputstring[3:-3]

character_key = read_encryption_details(key)
final_encryption = create_decryption(character_key)
plain_text = convert_cypher_to_plaintext(encrypted_text, final_encryption)

print(plain_text)

```

Output con la flag:

```

Enter plaintext: 2njlgkma2bv1i0v}221v19vuo19va2bv12{-5x
dsc{y0u_4r3_g00d_4t_wh4t_y0u_d0}

```

Behind Enemy Lines

One of our soldiers managed to intercept an secret message and some files. Help us decode it before the war is over.

Se nos entregan dos ficheros: `ciphertext.txt` e `instructions.pdf`.

El contenido de `ciphertext.txt` es el siguiente:

```
mckcu wiyqt jawul xedmi bclke ipdpp jdmvl gugks cggcq iiapb dkphr eymlv yrziv jzhmq lipab  
yrdbn suhpy wsqio tljot mrldl jzqmt qjmkn wahty oycpj ntvsh axhfn thrtu qtxfm ahiav xbqjt  
spuuf yyxvc qatli ewadf rksdd fhntl fbgjx arngn scwmk kweba rropy uoohj nciho rjolj fozny  
bxpdu zdqzf ljrvm nopcw ismtz exjql axues ioypx amqms jbyeb pyssp ehfv iof jilazhmfpyotp  
yzdz whykv xjkrb ejvcx qvusj fqgkn mbwli oihvo caqkz mvfoh wcrmp xoujk wcirt slxlf bwbhg  
ecwin tanav zvvrq aoqgv yndwt ieuxf wwqig qafan zjnyj bppmg eegmp gbkqx xlqwv ombdc tlidr  
rjtvd oefvj cvsg izlqf szmpj qdmoe rrcyt pveaa emijo njtus nvcoc iyagm imjzx ljcph xaqr1  
tkpsc vpwwn jyjxr skqsd brknj radag omfzk wjuyy jyslo ygdpo cprkn dcpzy ynffg eunzh fzkzx  
hetck lbunm qsxpu zbzof xoakd dovna dmxna ethux ewzfj fjcle ivjbq axkbs nxjwx aaesx hmvon  
zhnuj fkgzn wftrj jcihe hcknt ijfgw zidhn xlukp pwurl vyvpk idmck ybgfk velpb yomdz tivsy  
rdiyk kvggg jvwct sanep fuzfq j
```

El pdf con las instrucciones está protegido por contraseña así que utilicé una [herramienta online](#) para desbloquearlo:

GDSC CTF

Instructions are very clear. All challenges have to be saved in a format. The key for the following has been provided in the document.

Nothing here.

AgentAce

A simple vista no hay nada relevante, pero si seleccionamos todo el texto con `ctrl + A` se selecciona lo siguiente:

GDSC CTF

Instructions are very clear. All challenges have to be saved in a format. The key for the following has been provided in the document.

Nothing here.

AgentAce

Y al copiarlo y pegarlo se nos revela el siguiente texto:

Hidden Key Swiss KUKW 8H 7GIII 23 W 10JI 12L 21UII 17Q 14N-- you really thought, I wont hide it here?

El texto contiene la configuración de una máquina enigma modelo Swiss-K. Con otra [herramienta online](#) introduzco la configuración y el texto cifrado y descubro la flag en el texto en claro:

The screenshot shows the Cryptii interface with three main sections: Plaintext, Enigma machine, and Ciphertext.

- Plaintext:** Shows a long string of characters: ahcob zsek strhe lpfvi nzkry ukgfy lwiev jdqin nmeee nlhbf xhbtd ixish cjeeg txcyn rgtma npewy llouf qctpa deaffg vgbbr yprjx bafqv seuoa vanqt yccaz nhyre gzzl nxyan vogdx ncsfh xkthy jgdpr eicat ayxoaa khmwi jiwan oudsf ugadew oejar mtryo ffsns thzpn walgb mitif dbtfb aczvh vidgw wwdss zwfod lhcgw ggliy hgqvg hmfsv mxfwg yktli licca flagd scutur inglo vedme flagh qgvck ftxgk sjeqi eyvfo wiixv zaagt umvpx zhgcw eread vxelp lrnnv przrd zirgr lgnvz vrcvx zlshm rrxll xsogf dsznz vheqr sezom njimic pgnvs anbrh xxetrn xextv edlth cuwbb yptmt vskgw dlspr withz vgbkk juovl utpgx xalgm hcepo dbdyi tmykc tjrad izvap noewa uaynn syxhz jwans cpetz mfouq jpifm npddh rugxh jcapk svtho kurzw btatk1 keepz luima tveen gtgtt jhmjg ifqzr ulyld lulte twyld bvsvo dbcrw mnagg rtxqu hymkt evfda idrija eehpq covrp grpdk yilkh gfnel hskbg hnkoj pswz splb abilb ftrdg wdjtv gizba ilidg bdvnp gdisz ggmva hhkvj stlpn cdcvr rlakl ihact mpuzk cizep
- Enigma machine:** Set to Swiss-K model. Reflector: UKW (position H, ring +). Rotor 1: III (position W, ring +). Rotor 2: I (position L, ring +). Rotor 3: II (position Q, ring +). Foreign chars: Include Ignore.
- Ciphertext:** Shows the decrypted text: mckcu wiyat jawul xedmi bclke ipdpp jdmvl gugks cggcg iaiapb dkphr eymlv yrziv jzhmg lipab yrdbn subhp wsqio tljot mrldl jzgmt qinkt wahty oycki ntvsh axhfn thrtu qtxfm ahiax xbait spuuf vyxcv gatli ewadf rksdd fhntl fbgjx arngn scwmk kweba rropt uoohi ncioho rjoli fozny bxpnd zdqzf lirmw nocpw ismtz exjal axes joypx amqms jbyeb pyssp ehfv iof jilazhmfpyotu yvdz whykv xikrb ejvcx gusij fgkln mbwlj oihvo caqkz mvfol wcrmp xouik wcirt slxf bwbbg ecwin tanav zvvrq aoggv yndwt ieuxf wwqig qafan zinyi bppmg eegmp gbkqx xlqwv ombrd tlidr ritvd oeefvi c1vsg izlqf szmpz qdme rrcyt pveea enioj nitus nvcoo iyagn imizx licph xaqlr tkpsc vpwnn iyjxr skqsdf brknj radag onfzk wiuyv yyslo ygdp0 cprkn dcpxy yntfg eunzh fzkxz hetci lbnum qspu zbzof xoakd dovna dmnxn ethux ewzjf fjcle ivibq axkbs nxjwx aaezx hmvn zhnuy fkzgk wvfri icihc hcenktt iifgw zidhn xlukp pwurl vvvpl idmcx ybgfk velpb yomdz tivsy rdiyk kvggg jvwct sanep fuuzfg j

Flag:

dsc{turinglovedme}

