

Retos Hechos

- 1- El libro del Quijote
- 2- Collage
- 3- Suena bien
- 4- Capa de frutas
- 5- Mucho crypto
- 6- Ruta
- 7- Criptografía lejana
- 8- Cadena mágica
- 9- ¿Qué hace?
- 10- Análisis de correos
- 11- Calculator
- 12- Cofre

1- El libro del Quijote

Desde la creación del instituto, algunos alumnos se han encargado de la gestión del periódico local, "El noticiero", donde se intercambian mensajes y escriben artículos didácticos. En uno de los artículos alguien anónim@ ha lanzado un reto. Asegura haber escondido un mensaje en el libro más universal de la literatura, pero solo nos proporciona algunas coordenadas para que las mentes más brillantes del instituto puedan resolverlo.

¿Cuál es el mensaje secreto que ha enviado el alumno anónimo y ha escondido este anónim@?

Datos proporcionados:

Libro: "El quijote de la mancha" (adjunto)

Cada una de estas coordenadas pertenece a una palabra, juntalas mediante un guion bajo "_".

Si encuentras "palabra1" y "palabra2" la solución es flag{palabra1_palabra2}.

- 10:8:2 --> querer (página 10, línea 8, palabra 2)
- 23:10:1 --> saber (página 23 línea 10 palabra 1)
- 30:8:2 --> noticia (página 30, línea 8, palabra 2)
- 30:26:7 --> sobre (página 30 línea 26 palabra 7)
- 35:1:7 --> conocer (página 35, línea 1, palabra 7)
- 151:19:10 --> misterio (página 151 línea 19 palabra 10)

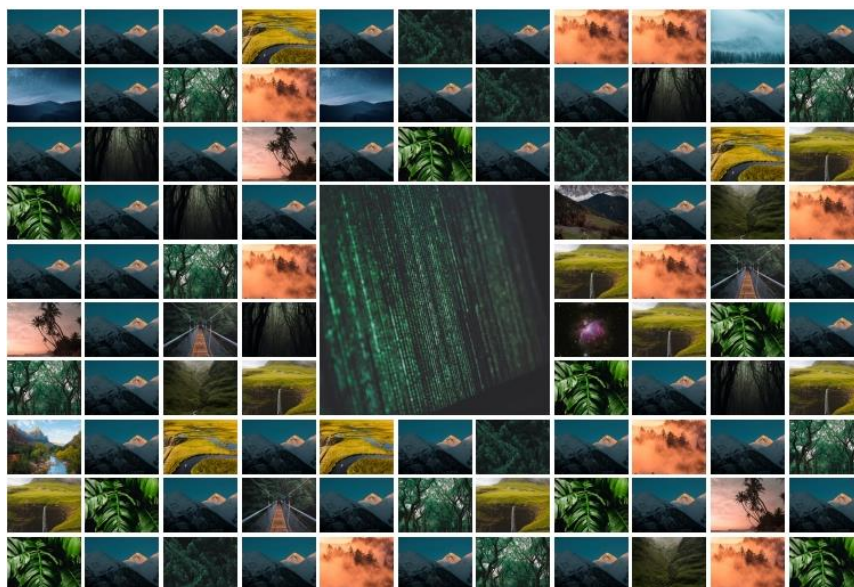
- 151:11:8 --> quien (página 151, línea 11, palabra 8)
- 152:11:5 --> hizo (página 152 línea 11 palabra 5)

SOLUCIÓN: {querer_saber_noticia_sobre_conocer_misterio_quien_hizo}.

2- Collage

La clase de manualidades e informática han preparado de manera conjunta un problema para los alumnos en el que se esconde un mensaje en un collage

¿Eres capaz de encontrar el mensaje oculto?



```

+ + + + + + + + + +
+ + + + + + + + + +
+ + + + + + + + + +
+ + + + - - - + + + +
+ + + + - - - + + + +
+ + + + - - - + + + +
+ + + + - - - + + + +
+ + + + + + + + + +
+ + + + + + + + + +
+ + + + + + + + + +

```

98 foto / 2 caracteres hexad. x foto = 49 caracteres:

flag{+++++++}

flag {} =

f l a g { }

66 6C 61 67 7B 7D

6 6 6 C 6 1 6 7 7 B 6

8 6 5 7 8 6 1 6 4 6 5

6 3 6 9 6 D 6 1 6 C 2

D 6 3 6 - - - F 6 E 7

6 6 5 7 - - - 2 7 4 6

9 6 4 3 - - - 0 2 D 6

5 6 E 2 - - - D 6 3 2

a 6 C 6 C 6 1 6 7 6 5

2 D 6 4 6 5 2 D 6 9 6

D 6 1 6 7 6 5 6 E 7 D

--> 666c61677b6

86578616465

63696d616c2

d636 f6e7

6657 2746

9643 02d6

56e2 d632

a6c6c616765

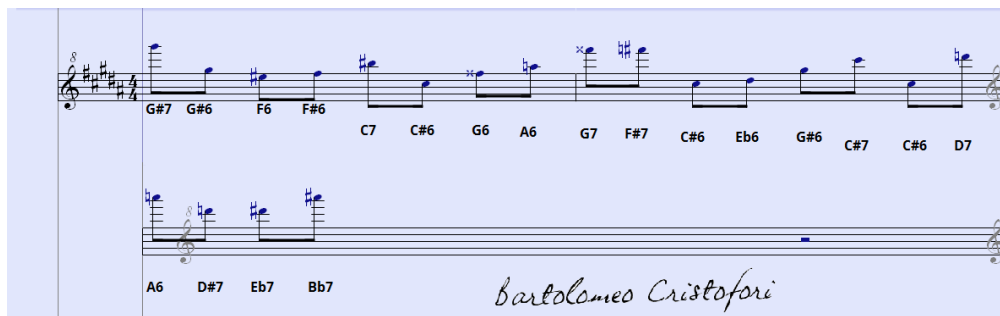
2d64652d696

d6167656e7d

666c61677b68657861646563696d616c2d636f6e766572746964302d656e2d632a6c
6c6167652d64652d696d6167656e7d

=>flag{hexadecimal-convertido-en-c*llage-de-imagen}

3- Suena bien



Los alumnos de un conservatorio musical han sido descubiertos copiando en un examen, la evidencia que el profesor ha descubierto (SuenaBien.png), aparentemente no se trata de una información confidencial, ¿Puedes observar el mensaje que estaban transmitiendo los alumnos?

Bartolomeo Cristofori --> inventó el piano --> partitura de piano

sol#6 = 84

sol#5 = 72

fa5 = 69

fa#5 = 70

do6 = 76

do#5 = 65

sol5 = 71

la5 = 73

sol6 = 83

fa#6 = 82

do#5 = 65

mib5 = 67

sol#5 = 72

do#6 = 77

do#5 = 65

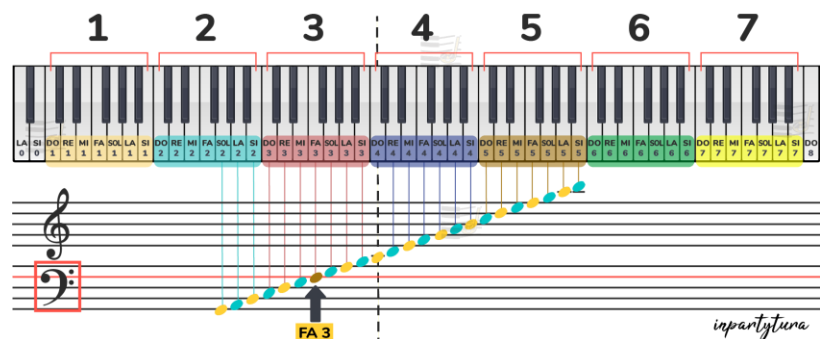
re6 = 78

la5 = 73

re#6 = 79

mib6 = 79

sib6 = 86



84 72 69 70 76 65 71 73 83 82 65 67 72 77 65 78 73 79 79 86

--> ASCII = 084 072 069 070 076 065 071 073 083 082 065 067 072 077 065 078 073
079 079 086

= THEFLAGISRACHMANIOOV

4- Capa de frutas

Enunciado:

¿Cuál es el sabor elegido para el nuevo zumo de una conocida marca? El fichero lemon.pdf parece tener la clave.

Pasos:

Primero nos enfrentamos a un pdf, que abriéndolo con un editor hexadecimal, vemos que no es un pdf al uso, sino que es un .tar. Por tanto, cambiamos la extensión .pdf a .tar, extraemos su contenido y descubrimos otro pdf con las mismas características, repetimos el proceso varias veces hasta que nos encontramos además un .txt que dice: "Muchas capas por delante. ¿Habrá alguna forma de automatizar este proceso?", teniendo esto, podemos optar por seguir repitiendo el proceso hasta que aparezca lo que buscamos o crear un script. En nuestro caso hemos hecho un script para el cmd de Windows:

```
@echo off
setlocal enabledelayedexpansion
:: Cambia la extensión de .pdf a .tar y descomprime los archivos .tar
:main_loop
set "changed=false"
:: Recorrer todas las carpetas y subcarpetas
for /r %%d in (.) do (
    pushd "%%d"
    :: Cambiar la extensión de .pdf a .tar
    for %%f in (*.pdf) do (
        echo Cambiando extensión de %%f a %%~nf.tar
        ren "%%f" "%%~nf.tar"
        set "changed=true"
    )
    :: Extraer los archivos .tar
    for %%f in (*.tar) do (
        echo Extrayendo %%f
        tar -xf "%%f"
        del "%%f"
    )
    popd
)
:: Si se hizo algún cambio, repetir el proceso
if "%changed%"=="true" (
    goto :main_loop
)
echo No more PDFs to process.
endlocal
pause
```

Este script permite darle a la tecla “enter” e ir ejecutándose en cada .pdf, terminando el proceso en un par de minutos. Cuando ya no quedan pdf para cambiar a .tar, descomprimir y volver a repetirlo, aparece un archivo .png, la flag:



5- Mucho crypto

¿Cuál es el sabor elegido para el nuevo zumo de una conocida marca? El fichero lemon.pdf parece tener la clave.

```
Vxoskx tobkr jk
iolxgju_OPACQ3XGTHYCM2JVLWWM65JYT4WM42RCSBCIG3RHUTVW====KA2ZYPPC
MKYZKSHLM4FYQTZMKA3MMPPCSEYZKSHLM4EYQTFBKA3JQPPCMWYZSSPLMOEI
QTFBKA3MQPPCSEYZQFW=KA3JSPPCSSYZSSPLME3YQT3IKA2JUPPAMSYZOTFLM
WIFYQTHZKA2JIPPAMKYZOSPLMW3YQTHXKA2JMPPBMWYZQTHLMA2IQTHZKA2ZOP
PBMWYZOTFLMWIFYQTPAKA2JIPPAMSYZOSPLMWIFYQTHZKA2JUPPAMKYZOSPLMA
2IQTPAKA2JMPPAMKYZQTHLMA2IQTPAKA2JM===KA3AO===
```

--> decode cesar:

```
Vxoskx tobkr jk
iolxgju_OPACQ3XGTHYCM2JVLWWM65JYT4WM42RCSBCIG3RHUTVW====KA2ZYPPC
MKYZKSHLM4FYQTZMKA3MMPPCSEYZKSHLM4EYQTFBKA3JQPPCMWYZSSPLMOEI
QTFBKA3MQPPCSEYZQFW=KA3JSPPCSSYZSSPLME3YQT3IKA2JUPPAMSYZOTFLM
WIFYQTHZKA2JIPPAMKYZOSPLMW3YQTHXKA2JMPPBMWYZQTHLMA2IQTHZKA2ZOP
PBMWYZOTFLMWIFYQTPAKA2JIPPAMSYZOSPLMWIFYQTHZKA2JUPPAMKYZOSPLMA
2IQTPAKA2JMPPAMKYZQTHLMA2IQTPAKA2JM===KA3AO===
```

----->>>>>

Primer nivel de

cifrado_IJUWK3RANBSWG2DPFQQG65DSN4QG42LWMVWCA3LBONPQ=====EU2TSJJ
WGESTEMBFG4ZSKNTGEU3GGJJWMYSTEMBFG4YSKNZVEU3DKJJWGQSTMMJFGIY
CKNZVEU3GKJJWMYSTKZQ=EU3DMJJWMMSTMMJFGY3SKN3CEU2DOJJUGMSTINZF
GQZSKNBTEU2DCJJUGESTIMJFGQ3SKNBREU2DGJJVGQSTKNBFGU2CKNBTEU2TIJJ
VGQSTINZFGQZSKNJUEU2DCJJUGMSTIMJFGQZSKNBTEU2DOJJUGESTIMJFGU2CK
NJUEU2DGJJUGESTKNBFGU2CKNJUEU2DG===EU3UI===

--> decode base32:

IJUWK3RANBSWG2DPFQQG65DSN4QG42LWMVWCA3LBONPQ=====EU2TSJJWGEST
EMBFG4ZSKNTGEU3GGJJWMYSTEMBFG4YSKNZVEU3DKJJWGQSTMMJFGIYCKNZV
EU3GKJJWMYSTKZQ=EU3DMJJWMMSTMMJFGY3SKN3CEU2DOJJUGMSTINZFGQZSK
NBTEU2DCJJUGESTIMJFGQ3SKNBREU2DGJJVGQSTKNBFGU2CKNBTEU2TIJJVGQST
INZFGQZSKNJUEU2DCJJUGMSTIMJFGQZSKNBTEU2DOJJUGESTIMJFGU2CKNJUEU2
DGJJUGESTKNBFGU2CKNJUEU2DG===EU3UI===

----->>>>>>

Bien hecho, otro nivel

mas_%59%61%20%73%6f%6c%6f%20%71%75%65%64%61%20%75%6e%6f%5f%66%6
c%61%67%7b%47%43%47%43%43%41%41%41%47%41%43%54%54%54%43%54%54
%47%43%54%41%43%41%43%43%47%41%41%54%54%43%41%54%54%54%43%7D

--> decode URL:

%59%61%20%73%6f%6c%6f%20%71%75%65%64%61%20%75%6e%6f%5f%66%6c%61
%67%7b%47%43%47%43%43%41%41%41%47%41%43%54%54%54%43%54%54%47%
43%54%41%43%41%43%43%47%41%41%54%54%43%41%54%54%54%43%7D

----->>>>>>

Ya solo queda uno_flag{GCGCCAAAGACTTTCTTGCTACACCGAATTCATTTC}

-> decode DNA cypher:

GCGCCAAAGACTTTCTTGCTACACCGAATTCATTTC

----->>>>>>

Much0 Crypt0

1-

VIEW Ciphertext

Vxoskx tobkr jk
ioLxgju_OPACQ3XGTHYCM2JVLWWM65
JYT4WM42RCSBCIG3RHUTVW===KA2Z
YPPCMKYZKSHLM4FYQTZMKA3MPPCSE
YZKSHLM4EYQTFBKA3JQPPCMWYZSSPL
MOEIQTFBKA3MQPPCSEYZQFW=KA3JSP
PCSSYZSSPLME3YQT3IKA2JUPPAMSYZ
OTFLMWFYQTHZKA2JIPAMKYZOSPLMW
3YQTHXKA2JMPPBMWYZQTHLMA2IQTHZ
KA2ZOPPBWYQZOTFLMWFYQTPAKA2JIP
PAMSYZOSPLMWFYQTHZKA2JUPPAMKYZ
OSPLMA2IQTPAKA2JMPPAMKYZQTHLMA
2IQTPAKA2JM===KA3AO===

ENCODE DECODE

Caesar cipher

SHIFT

6 a→g

ALPHABET

abcdefghijklmnopqrstuvwxyz

CASE STRATEGY

Maintain case

FOREIGN CHARS

Include Ignore

→ Decoded 368 chars

VIEW Plaintext

Primer nivel de
cifrado_IJUWK3RANBSWG2DPFQGG65
DSN4QG42LWMVWCA3LBONPQ===EU2T
SJJWGESTEMBF4ZSKNTGEU3GGJJWMY
STEMBFG4YSKNZVEU3DKJJWGQSTMMJF
GIYCKNZVEU3GKJJWMYSTKZQ=EU3DMJ
JWMMSTMMJFGY3SKN3CEU2DOJJUGMST
INZFGQZSKNBTEU2DCJJUGESTIMJFGQ
3SKNBREU2DGJJVGQSTKNBFGU2CKNBT
EU2TIJJVGQSTINZFGQZSKNJUEU2DCJ
JUGMSTIMJFGQZSKNBTEU2DOJJUGEST
IMJFGU2CKNJUEU2DGJJUGESTKNBFGU
2CKNJUEU2DG===EU3UI===

2-

Base32 Decode

Decode Base32 encoded data

IJUWK3RANBSWG2DPFQGG65DSN4QG42LWMVWCA3LBONPQ===EU2TSJJWGESTEMBF4ZSKNT
GEU3GGJJWMYSTEMBFG4YSKNZVEU3DKJJWGQSTMMJFGIYCKNZVEU3GKJJWMYSTKZQ=EU3DMJ
JWMMSTMMJFGY3SKN3CEU2DOJJUGMSTINZFGQZSKNBTEU2DCJJUGESTIMJFGQ3SKNBREU2DGJJ
VGQSTKNBFGU2CKNBTEU2TIJJVGQSTINZFGQZSKNJUEU2DCJJUGMSTIMJFGQZSKNBTEU2DOJJUG
ESTIMJFGU2CKNJUEU2DGJJUGESTKNBFGU2CKNJUEU2DG===EU3UI===

DECODE

Shareable url: <https://www.base64decode.net/base32-decode/oV>

Bien hecho, otro nivel

mas_%59%61%20%73%6f%6c%6f%20%71%75%65%64%61%20%75%6e%6f%5f%66%6c%61%67%7b%
47%43%47%43%43%41%41%41%47%41%43%54%54%43%54%54%47%43%54%41%43%41%43%
43%47%41%41%54%54%43%41%54%54%54%43%7D

3-

```
%59%61%20%73%6f%6c%6f%20%71%75%65%64%61%20%75%6e%6f%5f%66%6c%61%67%7b%47%43%47%43%43%41%41%41%47%41%43%54%54%54%43%54%54%47%43%54%41%43%41%43%43%47%41%41%54%54%43%41%54%54%54%43%7D
```

i For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

```
Ya solo queda uno_flag{GCGCCAAAGACTTTCTTGCTACACCGAATTCATTC}
```

4-

```
El número de serie del volumen es: 8ACS-LLA4

Directorio de C:\Users\anmab\Documentos\uni\TERCERO\2 CUATRIMESTRE\SEGURIDAD SR\RETOS\multicrypto\02-Descargables
29/05/2024 15:35 <DIR> .
29/05/2024 14:08 <DIR> ..
11/03/2021 20:13 368 MultiCrypto.txt
29/05/2024 15:35 1.220 python.py
29/05/2024 15:18 0 python.txt
3 archivos 1.588 bytes
2 dirs 227.240.177.664 bytes libres

C:\Users\anmab\Documentos\uni\TERCERO\2 CUATRIMESTRE\SEGURIDAD SR\RETOS\multicrypto\02-Descargables>python.py
File "C:\Users\anmab\Documentos\uni\TERCERO\2 CUATRIMESTRE\SEGURIDAD SR\RETOS\multicrypto\02-Descargables\python.py",
line 48
    'GTG : 'U',
      ^
SyntaxError: unterminated string literal (detected at line 48)

C:\Users\anmab\Documentos\uni\TERCERO\2 CUATRIMESTRE\SEGURIDAD SR\RETOS\multicrypto\02-Descargables>python.py
M u c h o C r y p t o

C:\Users\anmab\Documentos\uni\TERCERO\2 CUATRIMESTRE\SEGURIDAD SR\RETOS\multicrypto\02-Descargables>
```

6- Ruta

El abuelo de Francisco envió un e-mail desde Estados Unidos con las fotos de su viaje por la ruta 66. Para guardar la confidencialidad del contenido puso un archivo comprimido con password, y le indicó que en la imagen se encuentra la contraseña.

¿Puedes ayudar a Francisco a conocer la password?

| | | | | | | |
|---|---|----|---|---|---|---|
| H | N | I | E | U | F | O |
| I | C | S | I | G | A | T |
| C | R | \$ | O | D | L | O |
| I | E | M | S | A | F | M |
| M | I | A | R | E | E | N |
| O | B | L | E | T | H | E |
| S | L | A | R | U | T | A |

HNIEUFOICSIGATCR\$ODLOIEMSAFMMIAREENOBLETHESLARUTA

HICIMOS LA RUTA EN MOTO FUE INCREIBLE THE FLAG IS \$MAREADOS

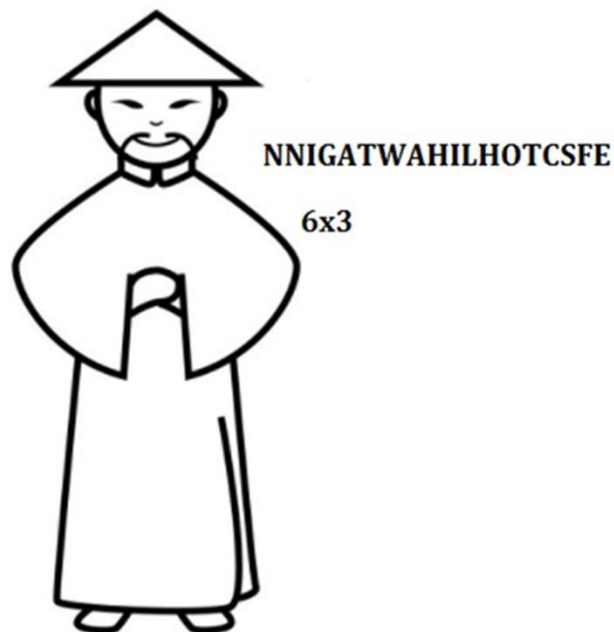


MAREADOS

Diego Espinosa, Manuel Neto, Lucía González y Paul Rodríguez

7- Criptografía lejana

Descifra el mensaje.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | ↑ | N | ↓ | I | ↑ | G | ↓ | A | ↑ | T | ↓ |
| W | | A | | H | | I | | L | | H | |
| O | | T | | C | | S | | F | | E | |

THE FLA GIS CHI NAT OWN -> **THE FLAG IS CHINA TOWN**

8- Cadena mágica

En clases de informática, el profesor está distribuyendo unos binarios que necesitan de una cadena mágica como parámetro para probar las habilidades de los alumnos

¿Puedes encontrar la cadena mágica que devuelve la flag?

Primero se abre el archivo que nos dan, lo estudiamos y se observa una cadena de caracteres: Introduce la cadena mágica como argumento `v7z7vw5rkcrcsiwchmjmgmp`
Lo siento, %s no es correcto.

Al introducir `v7z7vw5rkcrcsiwchmjmgmp` en el cmd se obtiene la cadena Lo siento, %s no es correcto.

En python se escribe estos códigos para poder descryptar la flag:

```
def rot47_decode(c):  
    if '!' <= c <= '~':  
        return chr(33 + ((ord(c) - 33 - 47) % 94))  
    return c
```

```
def decode_rot47(text):  
    return ''.join(rot47_decode(c) for c in text)
```

```
text = "v7z7vw5rkcrsciwchmjmgmp"  
rot47_decoded_text = decode_rot47(text)  
print(rot47_decoded_text)
```

```
def shift_char(c, shift):  
    if 'a' <= c <= 'z':  
        return chr((ord(c) - ord('a') + shift) % 26 + ord('a'))  
    elif 'A' <= c <= 'Z':  
        return chr((ord(c) - ord('A') + shift) % 26 + ord('A'))  
    elif '0' <= c <= '9':  
        return chr((ord(c) - ord('0') + shift) % 10 + ord('0'))  
    return c
```

```
def shift_text(text, shift):  
    return ''.join(shift_char(c, shift) for c in text)
```

```
final_decoded_text = shift_text(rot47_decoded_text, -4)  
print(final_decoded_text)
```

Y se obtiene:



```
(root@kali) ~ - [ /home/kali/Descargas/cadenamagica/02-Descargables ]  
$ ./flag r3v3rs1ng_no_es_d1f1c1l  
flag{r3v3rs1ng_no_es_d1f1c1l}
```

flag: r3v3rs1ng_no_es_d1f1c1l

9- ¿Qué hace?

En clase de informática te piden que compruebes qué hace el siguiente trozo de código. Aparentemente no realiza ninguna acción

¿Puedes hacerlo funcionar o extraer información de este?

Se obtiene:

```
var
_0x30de=['mZa1ntnUsLrKCKG','zwPLyW','s3jjsG','ota2ntjHELpsvg4','m21yBvviDG','nJC4ou
HKwfDzuq','zNvUyW','ndDzrLnir0S','E25VxW','odm3ndvzAuTPreS','mtm1m1rXCg5rCG','zM
XHzW','Aw9UFq','mtK2oti1y1jSBgzT','mti3sMnMCg9K','DxrHCG','mtnru2HKB2m','mNfkBe9
Nta','Bg9N','mta3nJG5wfrMEeHk','B2X2Aq'];var
_0x45c1=function(_0x41d678,_0x5378d3){_0x41d678=_0x41d678-(0x11*-0x79+0x2420*-
0x1+-0x2d07*-0x1);var
_0x6f67ce=_0x30de[_0x41d678];if(_0x45c1['iTqplC']===undefined){var
_0x1f4261=function(_0x1d2e0e){var
_0x18535e='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567
89+/-=';var _0x4d38ea="";for(var _0x1f83ef=-0x2*0x126a+-
0x17e2+0x3cb6,_0x40eab8,_0x4cab8c,_0x9867a0=-0xa*-0x52+0x5*-0x10+0x4a*-
0xa;_0x4cab8c=_0x1d2e0e['charAt'](_0x9867a0++);~_0x4cab8c&&(_0x40eab8=_0x1f83ef
%(0x12c6+0x52*-0xd+-0xe98)?_0x40eab8*(0xcb*-0x1c+0x1*-
0x803+0x1e77)+_0x4cab8c:_0x4cab8c,_0x1f83ef++%(-0x89*0xd+-0x12*-0xdc+-
0x57*0x19))?)_0x4d38ea+=String['fromCharCode']((0x23ce+0x1cd8+0x5*-
0xcbb&_0x40eab8>>(-(-0xde8+-0x4ae*-0x3+0x2*-0x10)*_0x1f83ef&0x211b*-
0x1+0x240d*0x1+0x44*-0xb)):-0x230*-0x5+-0x52*-0x1e+-
0x148c){_0x4cab8c=_0x18535e['indexOf'](_0x4cab8c);}return
_0x4d38ea;};_0x45c1['OpkxWb']=function(_0x1cd694){var
_0x19aed2=_0x1f4261(_0x1cd694);var _0x682b5b=[];for(var _0x10887a=-0x1f59+-
0x142f*0x1+0x11*0x308,_0x5616e7=_0x19aed2['length'];_0x10887a<_0x5616e7;_0x10887
a++){_0x682b5b+=%'+('00'+_0x19aed2['charCodeAt'](_0x10887a)['toString'](-0x695+-
0x1241*-0x2+0x37*-0x8b))['slice'](-(0x11f*-0x22+-0xd*0x11+0xcff*0x3));}return
decodeURIComponent(_0x682b5b);,_0x45c1['FVpRax']={},_0x45c1['iTqplC']=!![];var
_0x3cf3d8=_0x30de[-0x1b65+-
0x2118+0x3c7d],_0x541f8e=_0x41d678+_0x3cf3d8,_0x1d915e=_0x45c1['FVpRax'][_0x541f
8e];return
_0x1d915e===undefined?(_0x6f67ce=_0x45c1['OpkxWb'](_0x6f67ce),_0x45c1['FVpRax'][_0
x541f8e]=_0x6f67ce):_0x6f67ce=_0x1d915e,_0x6f67ce;};(function(_0x41153a,_0x2421b4){
var _0x2618b3=_0x45c1;while(![]){try{var
_0x5c9249=parseInt(_0x2618b3(0xf0))*parseInt(_0x2618b3(0xee))+
parseInt(_0x2618b3(0xe2))*parseInt(_0x2618b3(0xe4))+
parseInt(_0x2618b3(0xea))+parseInt(_0x2618b3(0xe1))*parseInt(_0x2618b3(0xe6))+
parseInt(_0x2618b3(0xe0))+parseInt(_0x2618b3(0xed))*parseInt(_0x2618b3(0xf2))+
parseInt(_0x2618b3(0xe7))*parseInt(_0x2618b3(0xeb));if(_0x5c9249===_0x2421b4)break;e
lse
_0x41153a['push'](_0x41153a['shift']());}catch(_0x44fefa){_0x41153a['push'](_0x41153a['shif
t']());}})(_0x30de,0xe1a6*-0x5+0x56ea2+-0x37*-0xaed));function _146154141147(){var
```

```

_0x52e3c3=_0x45c1,_0x4699b4={};_0x4699b4[_0x52e3c3(0xdf)+'D']=_0x52e3c3(0xe8)+_0
x52e3c3(0xe5)+_0x52e3c3(0xf1)+'des_'+_0x52e3c3(0xde)+_0x52e3c3(0xec)+'la_'+_0x52
e3c3(0xe3)+_0x52e3c3(0xe9);var
_0x2a9b51=_0x4699b4;console[_0x52e3c3(0xef)](_0x2a9b51[_0x52e3c3(0xdf)+'D']);}

```

Al desofuscarlo:

```

/** @type {Array} */
var _0x30de = ["mZa1ntnUsLrCKKG", "zwPLyW", "s3jjsG", "ota2ntjHELPSvg4",
"m21yBvviDG", "nJC4ouHKwfDzuq", "zNvUyW", "ndDzrLnir0S", "E25VxW",
"odm3ndvzAuTPreS", "mtm1m1rXCg5rCG", "zMXHzW", "Aw9UFq", "mtK2oti1y1jSBgzT",
"mti3sMnMCg9K", "DxrHCG", "mtnru2HKB2m", "mNfkBe9Nta", "Bg9N",
"mta3nJG5wfrMEeHk", "B2X2Aq"];
/**
 * @param {number} opt_attributes
 * @param {?} dataAndEvents
 * @return {?}
 */
var _0x45c1 = function(opt_attributes, dataAndEvents) {
  /** @type {number} */
  opt_attributes = opt_attributes - (17 * -121 + 9248 * -1 + -11527 * -1);
  var targetValue = _0x30de[opt_attributes];
  if (_0x45c1["iTqplC"] === undefined) {
    /**
     * @param {?} object
     * @return {?}
     */
    var getOwnPropertyNames = function(object) {
      /** @type {string} */
      var classNames =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/-=";
      /** @type {string} */
      var props = "";
      /** @type {number} */
      var bc = -2 * 4714 + -6114 + 15542;
      var bs;
      var buffer;
      /** @type {number} */
      var _0x9867a0 = -10 * -82 + 5 * -16 + 74 * -10;
      for (;buffer = object["charAt"](_0x9867a0++);~buffer && (bs = bc % (4806 + 82 * -13 + -
3736) ? bs * (203 * -28 + 1 * -2051 + 7799) + buffer : buffer, bc++ % (-137 * 13 + -18 * -220
+ -87 * 25)) ? props += String["fromCharCode"](9166 + 7384 + 5 * -3259 & bs >> (-(-3560 + -
1198 * -3 + 2 * -16) * bc & 8475 * -1 + 9229 * 1 + 68 * -11)) : -560 * -5 + -82 * -30 + -5260) {
        buffer = classNames["indexOf"](buffer);
      }
      return props;
    };
  }
};
/**

```

```

* @param {?} reqUrl
* @return {?}
*/
_0x45c1["OpkxWb"] = function(reqUrl) {
  var params = getOwnPropertyNames(reqUrl);
  /** @type {Array} */
  var sign = [];
  /** @type {number} */
  var i = -8025 + -5167 * 1 + 17 * 776;
  var l = params["length"];
  for (;i < l;i++) {
    sign += "%" + ("00" + params["charCodeAt"](i)["toString"](-1685 + -4673 * -2 + 55 * -
139))["slice"](-(287 * -34 + -13 * 17 + 3327 * 3));
  }
  return decodeURIComponent(sign);
};
_0x45c1["FVpRax"] = {};
/** @type {boolean} */
_0x45c1["iTqplC"] = !![];
}
var queueHooks = _0x30de[-7013 + -8472 + 15485];
var key = opt_attributes + queueHooks;
var val = _0x45c1["FVpRax"][key];
return val === undefined ? (targetValue = _0x45c1["OpkxWb"](targetValue),
_0x45c1["FVpRax"][key] = targetValue) : targetValue = val, targetValue;
};
(function(paths, radio) {
  /** @type {function (number, ?): ?} */
  var getter = _0x45c1;
  for (;!![];) {
    try {
      /** @type {number} */
      var value = parseInt(getter(240)) * parseInt(getter(238)) + -parseInt(getter(226)) * -
parseInt(getter(228)) + -parseInt(getter(234)) + parseInt(getter(225)) * -parseInt(getter(230))
+ -parseInt(getter(224)) + -parseInt(getter(237)) * -parseInt(getter(242)) + -
parseInt(getter(231)) * parseInt(getter(235));
      if (value === radio) {
        break;
      } else {
        paths["push"](paths["shift"]());
      }
    } catch (_0x44fefa) {
      paths["push"](paths["shift"]());
    }
  }
})(_0x30de, 57766 * -5 + 356002 + -55 * -2797);
/**
* @return {undefined}

```



```

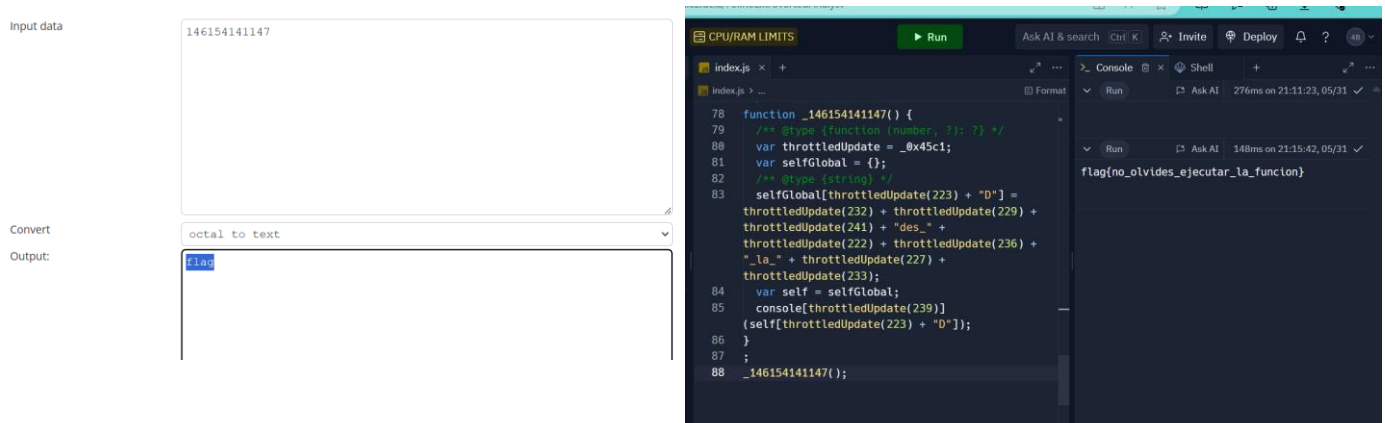
*/
function _146154141147() {
  /** @type {function (number, ?): ?} */
  var throttledUpdate = _0x45c1;
  var selfGlobal = {};
  /** @type {string} */
  selfGlobal[throttledUpdate(223) + "D"] = throttledUpdate(232) + throttledUpdate(229) +
  throttledUpdate(241) + "des_" + throttledUpdate(222) + throttledUpdate(236) + "_la_" +
  throttledUpdate(227) + throttledUpdate(233);
  var self = selfGlobal;
  console[throttledUpdate(239)](self[throttledUpdate(223) + "D"]);
}
;

```

Se observa que la función `function _146154141147()` no se ejecuta nunca, si la decodificamos en octal tenemos: flag

Por tanto ejecutamos llamamos a esa función para obtener la flag:

Obtenemos: `flag{no_olvides_ejecutar_la_funcion}`



10- Análisis de correos

Se ha logrado acceder al equipo de D. Furioso y se ha realizado una extracción de su carpeta de correo electrónico. Examina las comunicaciones y encuentra información acerca de un percance sufrido por D. Furioso. Indica la matrícula del coche de D.Furioso.

Entre los archivos descargables se encuentra una carpeta de nombre `thundebird`, dentro de la cual vemos `ccnsqul8.default`, `Crash Reports` y `profiles.ini`, entramos en `ccnsqul8.default` y descubrimos gran cantidad de archivos y carpetas, de entre todos ellos abrimos `Mail`, dentro de la cual aparecen `Local Folders` y `pop-mail.outlook.com`, decidimos abrir `pop-mail.outlook.com` y se muestran varios archivos, escogemos el que se llama `Inbox` y lo abrimos en el notepad, si lo estudiamos con un poco de atención, entorno a la mitad del archivo encontramos:

Matricula: C047057-R

11- Calculator

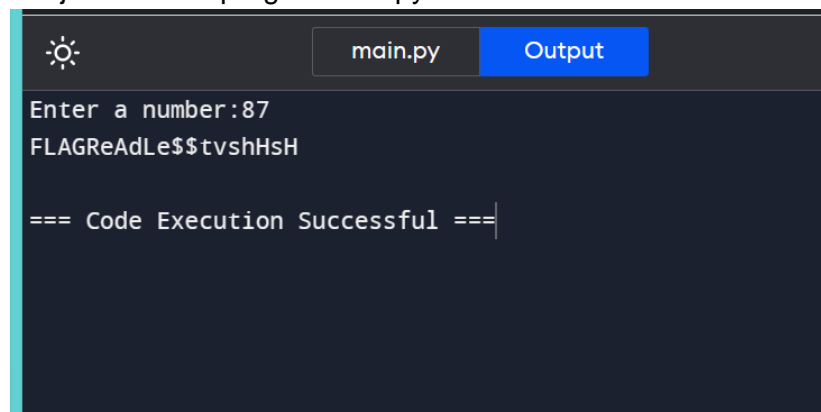
Un amigo nos ha pasado el código de las imágenes y cuando lo ejecutamos nos pide un número. Nuestro amigo no nos ha dado el número, que es necesario para que el programa funcione bien.

Se obtienen estas dos imágenes

```
1 number = input("Enter a number:")
2
3 a=200
4 b=0
5
6 a=a/2
7 a=a-7
8 a+=1
9
10 b+=1
11 b=b*70
12 b=b+12
13 b-=1
14
15 m="FLAG:"
16 q="MUcH"
17 w="Le$$"
18 e="shHsH"
19 r="ReAd"
20 t="wAtcH"
21 y="NoT"
22 u="tV"
23 i="f1Lm"
24 o="B0oK"
25 p="d0"
26 s="pWd"
27 d="C0mPuTeR"
28 f="12345"
29 g="54321"
30 h="AbCdE"
31 j="..."
32
33 if b < number < a:
34     if number%2 == 0:
35         if number == 88:
36             print j+f+d+u+e
37         elif number == 84:
38             print s+r+o+j+g
39         elif number == 90:
40             print t+h+d+i+y
41         else:
42             print q+w+e+r+t
43     else:
44         if number == 93:
45             print y+s+h+o+e
46         elif number == 87:
47             print m+r+w+u+e
48         elif number == 91:
49             print t+h+d+i+y
50         else:
51             print q+w+e+r+t
52 else:
53     print "Nice try, but you have failed :("

```

Al ejecutar este programa en python se obtiene:



```

Enter a number:87
FLAGReAdLe$$tvshHsH

=== Code Execution Successful ===

```

flag: FLAGReAdLe\$\$tvshHsH

12- Cofre

Un amigo nos ha pasado un fichero cofre.pyc y partes del código. Cuando lo ejecutamos nos pide una contraseña. Nuestro amigo no nos ha dado la contraseña, que es necesaria para que el programa funcione bien. Cuidado, que es un poco troll.

```
def main():
    var1 = raw_input('Valor 1: ')
    print var1
    var2 = raw_input('Valor 2: ')
    print var2

    if primeraClave(var1)== True:
        if segundaClave(var1,var2)==True:
            dameflag()
```

```
def primeraClave(clave):
    if clave == '1':
        correcto=True
        print"Has elegido el numero 1"
    elif clave == '2':
        correcto=True
        print"Has elegido el numero 2"
    elif clave == '3':
        correcto=True
        print"Has elegido el numero 3"
    elif clave == '4':
        correcto=True
        print"Has elegido el numero 4"
    elif clave == '5':
        correcto=True
        print"Has elegido el numero 5"
    else:
        print('No has pasado el primer control.')
        correcto =False
    return correcto
```

```
def segundaClave(clave,clave2):
    if len(clave2) != (2*int(clave)):
        print "No has pasado el segundo control";
        correcto2= False
    else:
        print('Segundo control pasado')
        correcto2= True
    return correcto2
```

```
def dameflag():
    sleep_time = random.randrange(1,5)
    word = "jaja"

    while len(word) != 7:
        #time.sleep(sleep_time)
        print "Buu!", word
        word = raw_input("Introduce la password:")
        print word

    if word.startswith('T'):
        if word.count(chr(97)) == 6:
            res = 'flag:00000000000'
            word.endswith('d')
            res = 'no ' + res + 'Python'
            print res
        elif word.count(chr(97)) == 3:
            res = 'flag: 43 6c 61 72 6f 51 75 65 53 69 47 75 61 70 69'
            res = ''
            if word.endswith('d'):
                print "Ok, gracias"
                res = '1346-000000000000'
                list1 = list(res)
                list1[13] =
                list1[16] =
                list1[12] =
                list1[11] =
                list1[5] =
                list1[6] =
                list1[15] =
                list1[7] =
                list1[8] =
                list1[9] =
                list1[10] =
                list1[14] =
                res = ''.join(list1)
                res = res + 'Python'
                print "The flag is: " + res
            else:
                res = 'flag:00000000000'
                res = 'no ' + res + 'Python'
                now = datetime.now()
                mm = str(now.month)
                dd = str(now.day)
                yyyy = str(now.year)
                print dd + '/' + mm + '/' + yyyy
            print "Bye!"
```



Observando el código se llega a la conclusión de que el primer valor debe ser un número entre 1 y 5. Si se decodifica el valor de res: 43 6c 61 72 6f 51 75 65 53 69 47 75 61 70 69 = ClaroQueSiGuapi. También se concluye que el segundo valor debe tener como longitud el doble del primer valor, la palabra debe tener 7 caracteres, empezar por 'T', el carácter 'a' debe estar repetido 3 veces. T...aaa

Ejecutando el código e introduciendo los valores 1, 12, Teniaaaa se consigue la flag:

1346-EILaberinToDePython