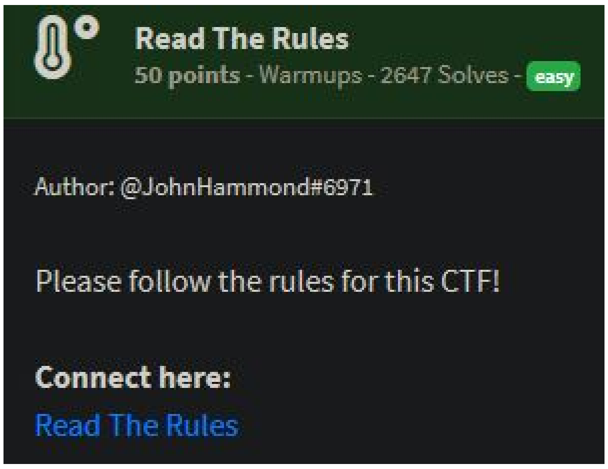
Writeup for NahamCon CTF 2021

Writeup for Wargames.my 2020

Category: Warmup - Read The Rules



Read The Rules

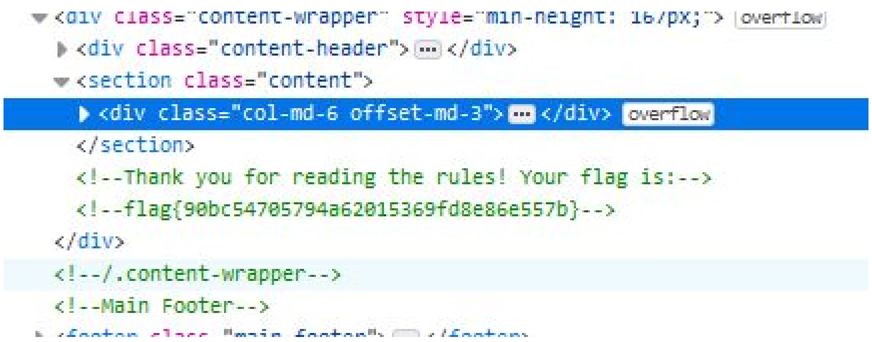
50 points - Warmups - 2647 Solves - easy

Author: @JohnHammond#6971

Please follow the rules for this CTF!

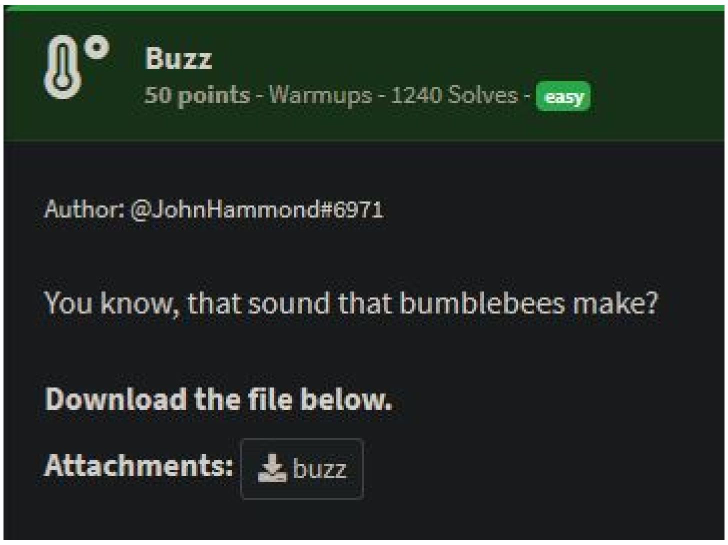
Connect here:
Read The Rules

It in the source code of the CTF rule page



```
▼<div class="content-wrapper" style="min-height: 167px;"> overflow
  ▶<div class="content-header">…</div>
  ▼<section class="content">
    ▶<div class="col-md-6 offset-md-3">…</div> overflow
  </section>
  <!--Thank you for reading the rules! Your flag is:-->
  <!--flag{90bc54705794a62015369fd8e86e557b}-->
 </div>
 <!--/.content-wrapper-->
 <!--Main Footer-->
```

Category: Warmup - Buzz



Buzz

50 points - Warmups - 1240 Solves - easy

Author: @JohnHammond#6971

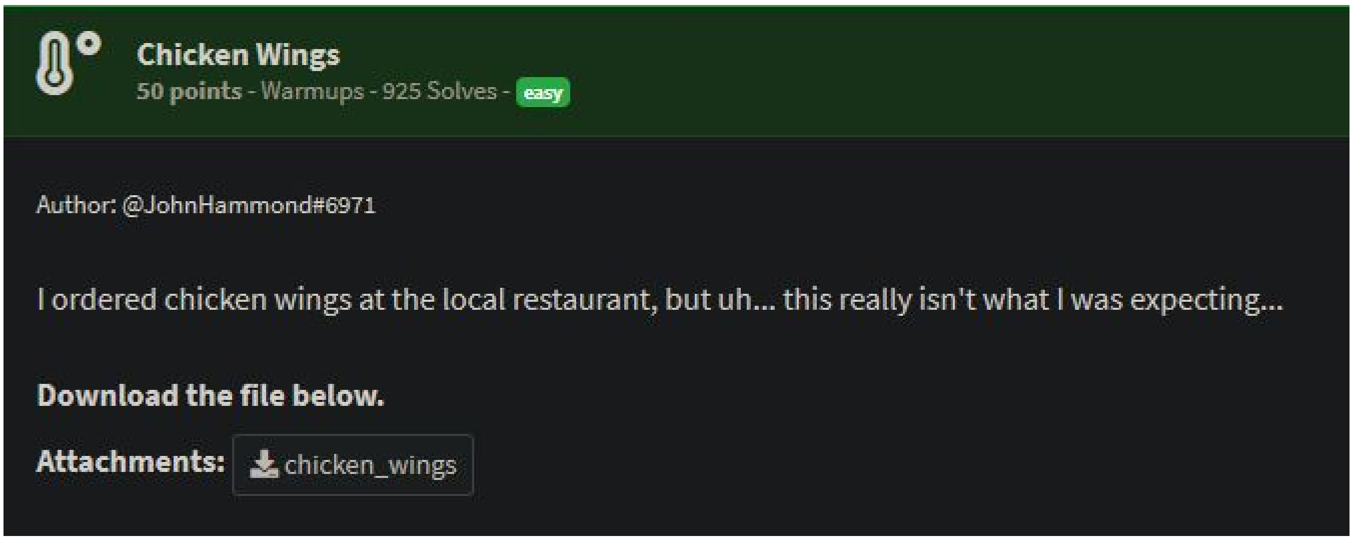
You know, that sound that bumblebees make?

Download the file below.

Attachments: buzz

Open file using HxD show file signature of `1F 9D` which is tar zip file. Rename the file into a zip file such as `.zip` or `.z` allow it to be extracted and reveal a file containing the flag that can be open with any notepad.

Category: Warmup - Chicken Wings


Chicken Wings

50 points - Warmups - 925 Solves - easy

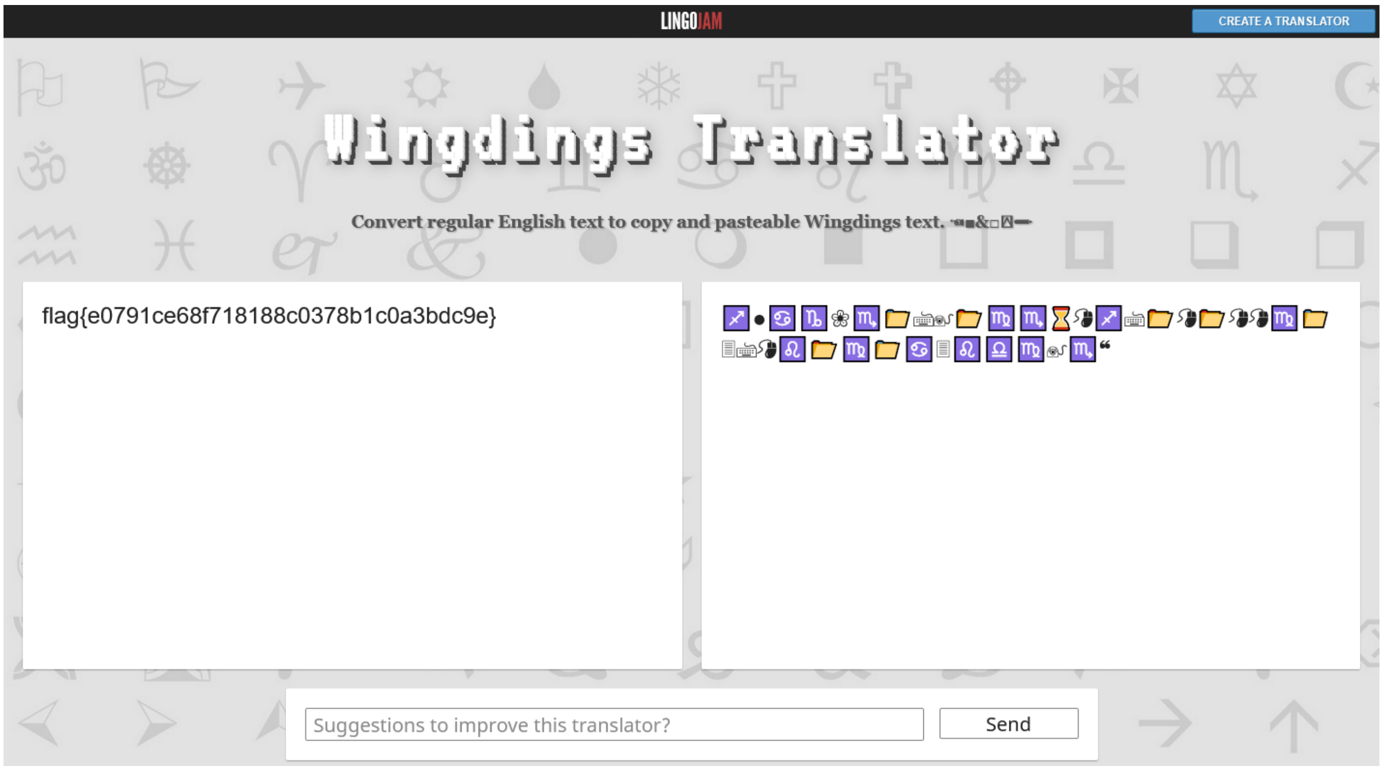
Author: @JohnHammond#6971

I ordered chicken wings at the local restaurant, but uh... this really isn't what I was expecting...

Download the file below.

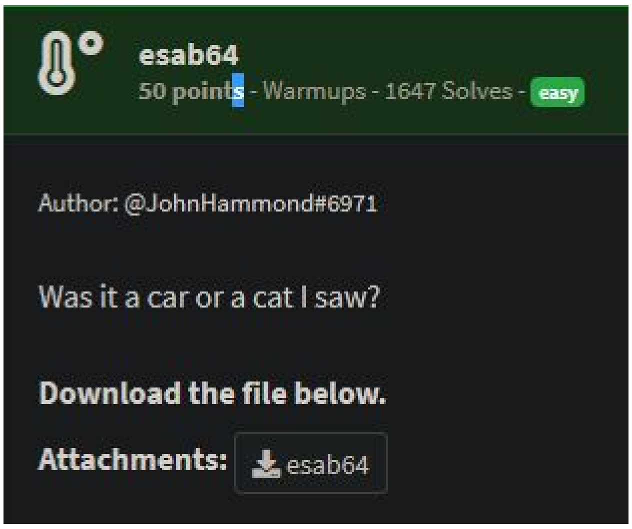
Attachments: chicken_wings

Open file in notepad shows a string of symbols, this string are known as wingdings. Use online [Wingdings Translator](#).


LINGOJAM
CREATE A TRANSLATOR
Wingdings Translator
Convert regular English text to copy and pasteable Wingdings text.
Suggestions to improve this translator?
Send

```
flag{e0791ce68f718188c0378b1c0a3bdc9e}
```

Category: Warmup - esab64



esab64
50 points - Warmups - 1647 Solves - easy

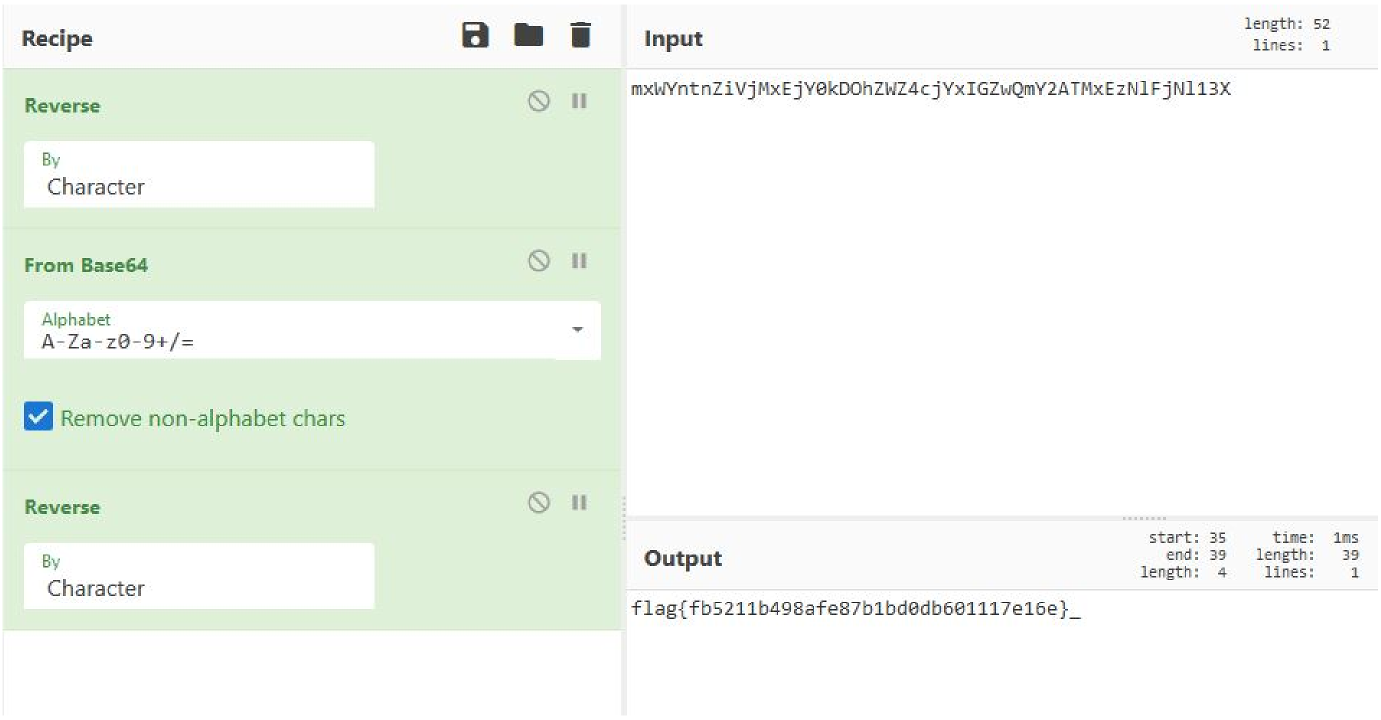
Author: @JohnHammond#6971

Was it a car or a cat I saw?

Download the file below.

Attachments: esab64

Open file in notepad shows a string. Name of the question give a hint on converting the strings through base64 and reverse the string.



Recipe

Reverse
By: Character

From Base64
Alphabet: A-Za-z0-9+/=
☑ Remove non-alphabet chars

Reverse
By: Character

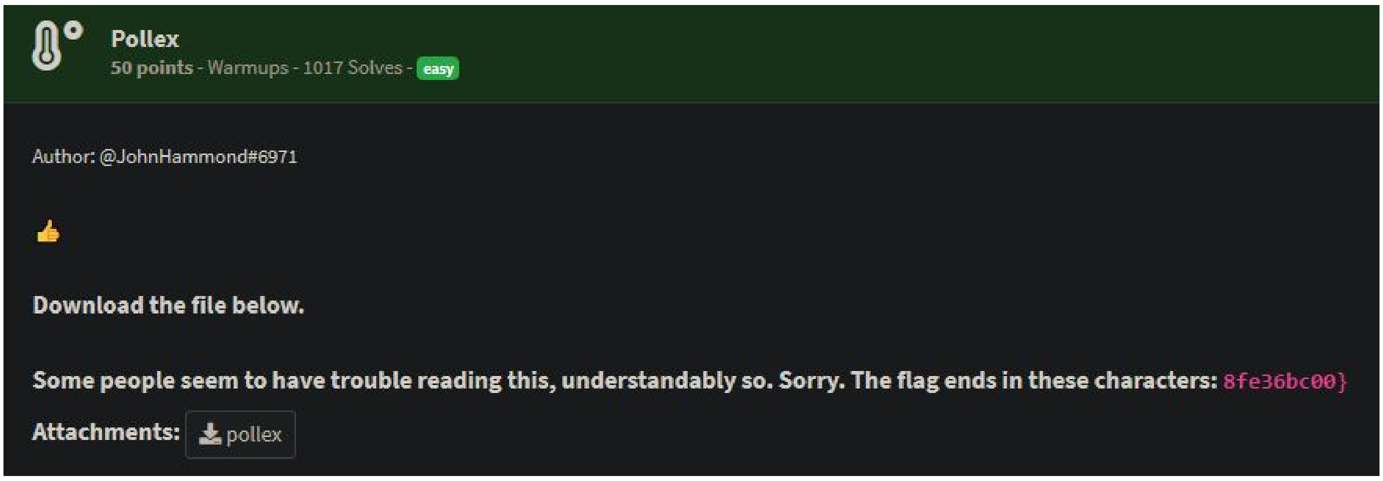
Input

```
length: 52
 lines:  1
mxWYntnZiVjMxEjY0kDOhZWZ4cjYxIGZwQmY2ATMxEzNlFjNl13X
```

Output

```
start: 35     time:  1ms
  end: 39   length:   39
length:  4   lines:    1
flag{fb5211b498afe87b1bd0db601117e16e}_
```

Category: Warmup - Pollex



Pollex
50 points - Warmups - 1017 Solves - easy

Author: @JohnHammond#6971

👍

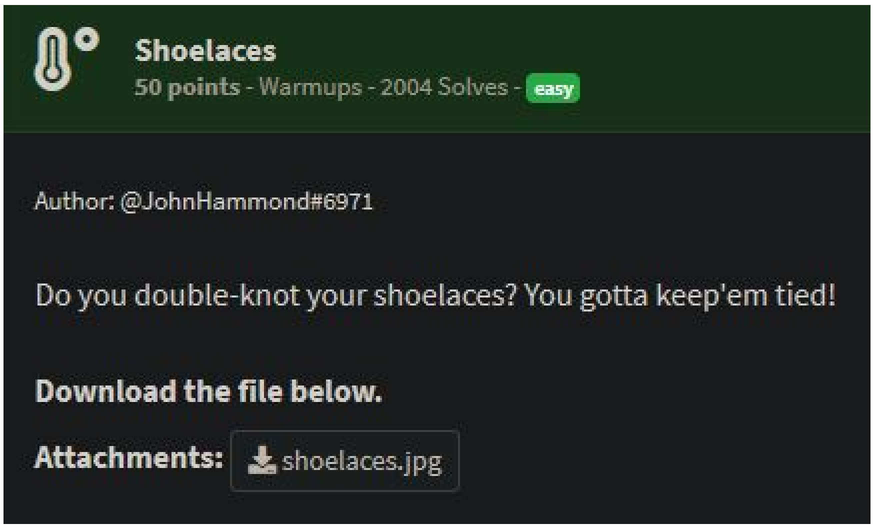
Download the file below.

Some people seem to have trouble reading this, understandably so. Sorry. The flag ends in these characters: `8fe36bc00}`

Attachments: pollex

The flag is actually on the thumbnail of the image. Zoom in on the thumbnail to read the flag.

Category: Warmup - Shoelaces


Shoelaces

50 points - Warmups - 2004 Solves - easy

Author: @JohnHammond#6971

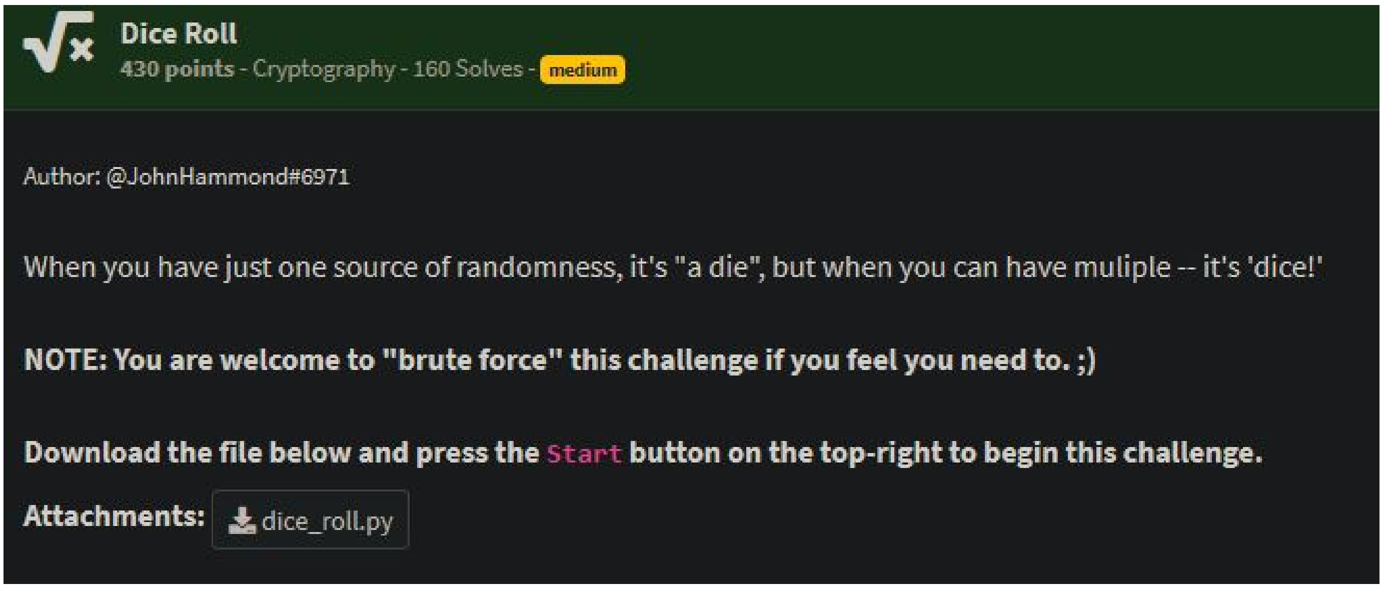
Do you double-knot your shoelaces? You gotta keep'em tied!

Download the file below.

Attachments: shoelaces.jpg

Open file in notepad and search `flag{`.

Category: Cryptography - Dice Roll


Dice Roll

430 points - Cryptography - 160 Solves - medium

Author: @JohnHammond#6971

When you have just one source of randomness, it's "a die", but when you can have muliple -- it's 'dice!'

NOTE: You are welcome to "brute force" this challenge if you feel you need to. ;)

Download the file below and press the `Start` button on the top-right to begin this challenge.

Attachments: dice_roll.py

Below is the source code

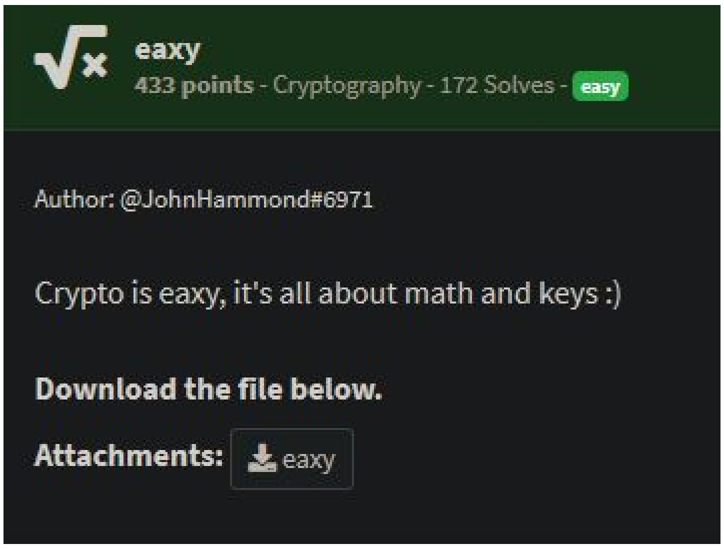
```
import random
import os

banner = """
              _______
   ______    | .   . |\\
  /     /\\   |   .   |.\\
 /  '  /  \\  | .   . |.'|
/_____/. . \\ |_______|.'|
\\ . . \\    /  \\ ' .  \\'|
 \\ . . \\  /    \\____'__\\|
  \\______\\/

      D I C E   R O L L
"""

menu = """
0. Info
1. Shake the dice
2. Roll the dice (practice)
3. Guess the dice (test)
"""

dice_bits = 32
flag = open('flag.txt').read()
print(banner)

while 1:
    print(menu)

    try:
        entered = int(input('> '))
    except ValueError:
        print("ERROR: Please select a menu option")
        continue

    if entered not in [0, 1, 2, 3]:
        print("ERROR: Please select a menu option")
        continue

    if entered == 0:
        print("Our dice are loaded with a whopping 32 bits of
randomness!")
        continue

    if entered == 1:
        print("Shaking all the dice...")
        random.seed(os.urandom(dice_bits))
        continue

    if entered == 2:
        print("Rolling the dice... the sum was:")
        print(random.getrandbits(dice_bits))
        continue

    if entered == 3:
        print("Guess the dice roll to win a flag! What will the sum total
be?")
        try:
            guess = int(input('> '))
        except ValueError:
            print("ERROR: Please enter a valid number!")
            continue

        total = random.getrandbits(dice_bits)
        if guess == total:
            print("HOLY COW! YOU GUESSED IT RIGHT! Congratulations! Here
is your flag:")
            print(flag)
        else:
            print("No, sorry, that was not correct... the sum total was:")
            print(total)

        continue
```

There is menu to reset the `random.seed()` value and get the value of the `random.getrandbits()` as much as we can. Therefore, we can predict the next state or number when given enough value of the generated number. More information here

I use Mersenne Twister Predictor to predict the number.

```
import random
from pwn import *
from mt19937predictor import MT19937Predictor

predictor = MT19937Predictor()
r = remote('challenge.nahamcon.com', 31784)

t = r.recvuntil("> ")
#r.sendlineafter("> ", str(1))
r.sendline(str(1))
i = 1
for _ in range(624):
    print(i)
    t = r.recvuntil("> ")
    r.sendline(str(2))
    t = r.recvuntil(":")
    print(t)
    t = r.recvline().decode('UTF-8').rstrip()
    t = r.recvline().decode('UTF-8').rstrip()
    print(t)
    predictor.setrandbits(int(t), 32)
    i+=1

t = r.recvuntil("> ")
r.sendline(str(3))
t = r.recvuntil("> ")
r.sendline(str(predictor.getrandbits(32)))
t = r.recvuntil(":")
print(t)
t = r.recvline().decode('UTF-8').rstrip()
t = r.recvline().decode('UTF-8').rstrip()
print(t)
```

Category: Cryptography - eaxy



eaxy

433 points - Cryptography - 172 Solves - easy

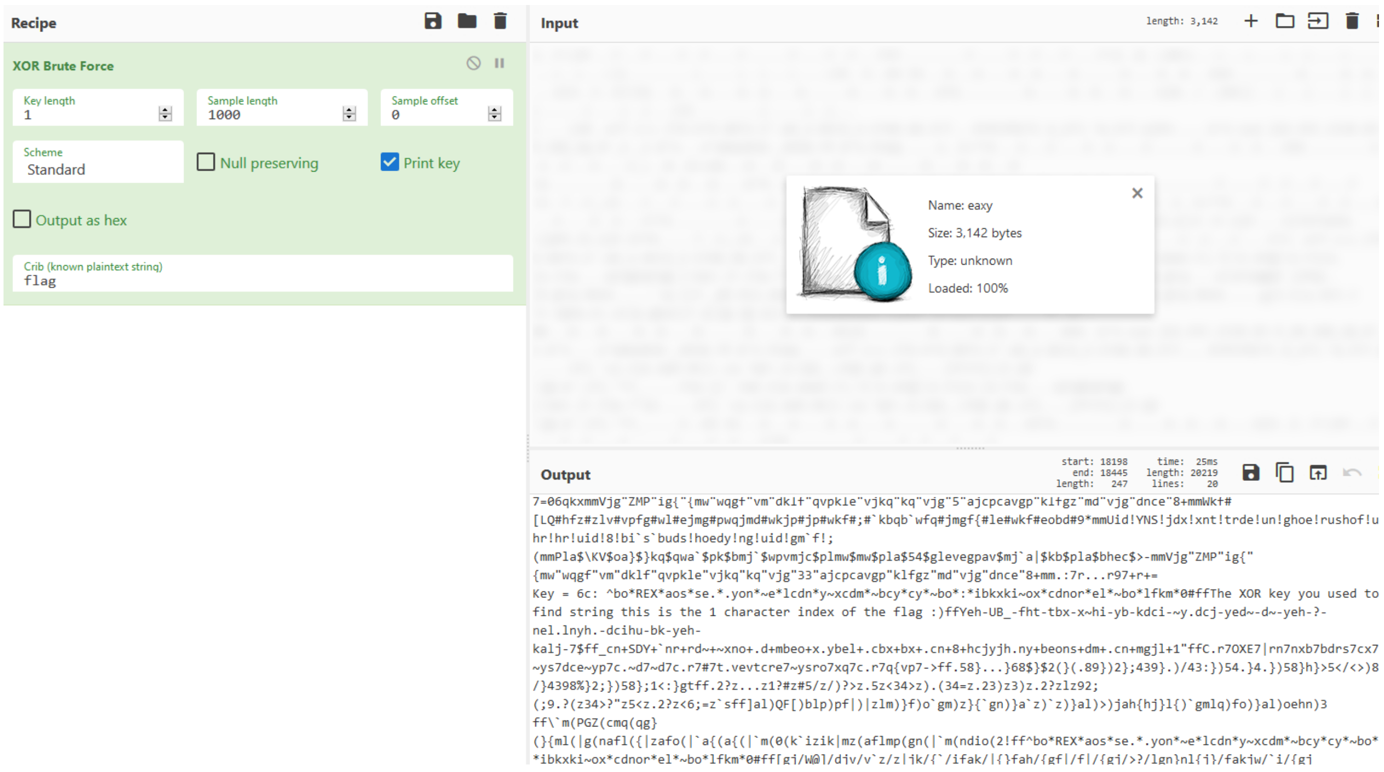
Author: @JohnHammond#6971

Crypto is eaxy, it's all about math and keys :)

Download the file below.

Attachments: eaxy

File consist of garbage string, that have no file signature whatsover. Therefore, we try to XOR the file using cyberchef XOR Brute Force



Recipe

XOR Brute Force

Key length: 1
Sample length: 1000
Sample offset: 0
Scheme: Standard
Null preserving: ☐
Print key: ✓
Output as hex: ☐
Crib (known plaintext string): flag

Input length: 3,142

Name: eaxy
Size: 3,142 bytes
Type: unknown
Loaded: 100%

Output

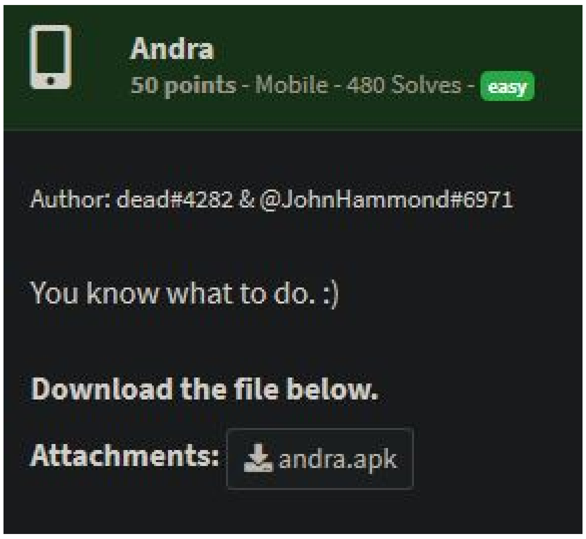
start: 18198 end: 18445 length: 247 time: 25ms length: 20219 lines: 20

```
7=06qkxmmVjg"ZMP"ig{"{mw"wqgt"vm"dklt"qvpkle"vjkq"kq"vjg"5"ajcpcavgp"kltgz"md"vjg"dnce"8+mmWkt#
[LQ#hfz#zlv#vpfg#wl#ejmg#pwqjmd#wkjp#jp#wkf#;#`kbqb`wfq#jmgf{#le#wkf#eobd#9*mmUid!YNS!jdx!xnt!trde!un!ghoe!rushof!u
hr!hr!uid!8!bi`s`buds!hoedy!ng!uid!gm`f!;
(mmPla$\KV$oa}$}kq$qwa`$pk$bmj`$wpvmjc$plmw$mw$pla$54$glevegpav$mj`a|$kb$pla$bhec$>-mmVjg"ZMP"ig{"
{mw"wqgf"vm"dklf"qvpkle"vjkq"kq"vjg"33"ajcpcavgp"klfgz"md"vjg"dnce"8+mm.:7r...r97+r+=
Key = 6c: ^bo*REX*aos*se.*.yon*~e*lcdn*y~xcdm*~bcy*cy*~bo*:*ibkxki~ox*cdnor*el*~bo*lfkm*0#ffThe XOR key you used to
find string this is the 1 character index of the flag :)ffYeh-UB_-fht-tbx-x~hi-yb-kdci-~y.dcj-yed~-d~-yeh-?-
nel.lnyh.-dcihu-bk-yeh-
kalj-7$ff_cn+SDY+`nr+rd~+~xno+.d+mbeo+x.ybel+.cbx+bx+.cn+8+hcjyjh.ny+beons+dm+.cn+mgjl+1"ffC.r7OXE7|rn7nxb7bdrs7cx7
~ys7dce~yp7c.~d7~d7c.r7#7t.vevtcre7~ysro7xq7c.r7q{vp7->ff.58}...}68$}$2(}(.89})2};439}.)/43:})54.}4.})58}h}>5</<>)8
/}4398%}2;})58};1<:}gtff.2?z...z1?#z#5/z/)?>z.5z<34>z).(34=z.23)z3)z.2?zlz92;
(;9.?(z34>?"z5<z.2?z<6;=z`sff]al)QF[)blp)pf|)|zlm)}f)o`gm)z}{`gn)}a`z)`z)}al)>)jah{hj}l{)`gmlq)fo)}al)oehn)3
ff\`m(PGZ(cmq(qg}
(}{ml(|g(nafl({|zafo(|`a{(a{(|`m(0(k`izik|mz(aflmp(gn(|`m(ndio(2!ff^bo*REX*aos*se.*.yon*~e*lcdn*y~xcdm*~bcy*cy*~bo*
*ibkxki~ox*cdnor*el*~bo*lfkm*0#ff[gi/W@]/div/v`z/z|ik/{`/ifak/|{}fah/{gf|/f|/{gi/>?/lgn}nl{i}/fakjw/`i/{gi
```

We figure it out that the key used for XOR will show this string `The XOR key you used to find string this is the 3 character index of the flag :)`. This mean that a singular char is used as the key for the XOR. Because of this we prepare an array of character that will be used for the key. Then we filter out the readable string in output and get the index number into flag output.

```python
key =
['0','1','2','3','4','5','6','7','8','9','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',
'{', '}']

flag = ["-"]*38
for keys in key:
    b = bytearray(open('eaxy', 'rb').read())
    for i in range(len(b)):
        b[i] ^= int(hex(ord(keys)),16)
    if "The XOR key you used" in str(b):
        print("Character ",keys)
        for i in str(b).split("this is the ")[1:]:
            print(i[0:2])
            flag[int(i[0:2])] = str(keys)
    print()

print("".join(flag))
```

Category: Mobile - Andra


Andra
50 points - Mobile - 480 Solves - easy

Author: dead#4282 & @JohnHammond#6971

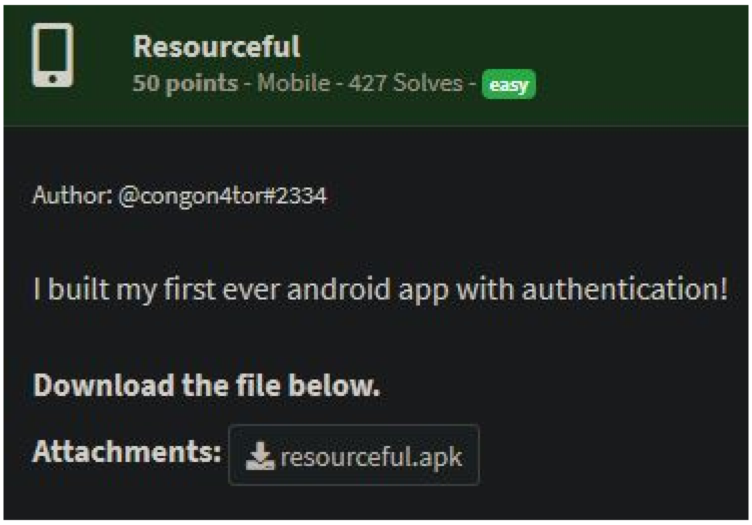
You know what to do. :)

Download the file below.

Attachments: andra.apk

Use tool `apktool` to decode the `.apk`. After that perform search on the apk folder for string `flag{`. The flag is in `\andra\res\layout\activity_flag.xml`

Category: Mobile - Resourceful


Resourceful
50 points - Mobile - 427 Solves - easy

Author: @congon4tor#2334

I built my first ever android app with authentication!

Download the file below.

Attachments: resourceful.apk

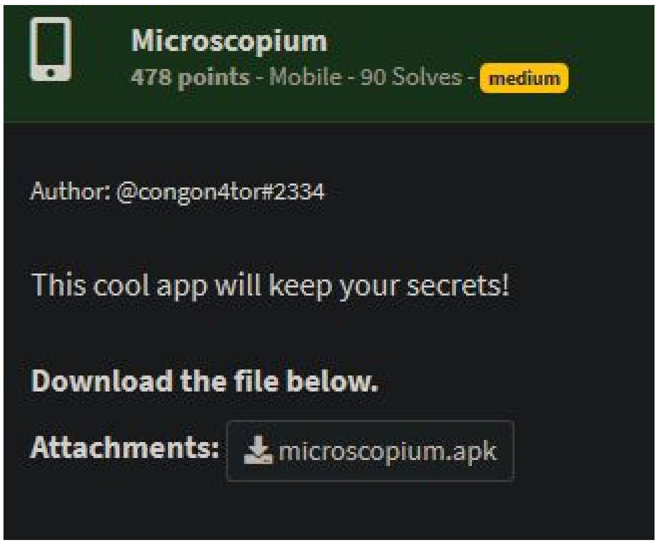
Use tool `dex2jar` to convert the `.apk` into `.jar`. Then look into `MainActivity.class` for the password

```
public class MainActivity extends AppCompatActivity {
  protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131361821);
    final EditText password = (EditText)findViewById(2131165351);
    ((Button)findViewById(2131165404)).setOnClickListener(new
View.OnClickListener() {
          public void onClick(View param1View) {
            if
(password.getText().toString().equals("sUp3R_S3cRe7_P4s5w0Rd")) {
              Intent intent = new Intent((Context)MainActivity.this,
FlagActivity.class);
```

```
                MainActivity.this.startActivity(intent);
              } else {
                Toast.makeText(MainActivity.this.getBaseContext(), "Error:
Incorrect password", 1).show();
              }
            }
          });
    }
}
```

Install the apk and use the password will reveal the flag.

Category: Mobile - Microscopium


Microscopium
478 points - Mobile - 90 Solves - medium

Author: @congon4tor#2334

This cool app will keep your secrets!

Download the file below.

Attachments: microscopium.apk

Use tool `apktool` to decode the `.apk`. It shows that it is using react native which bundle up some file in assets folder `index.android.bundle`. Using React Native Decompiler to decompile the file revealing large number of `.js` file.

The file `400.js` contains the function for the password.

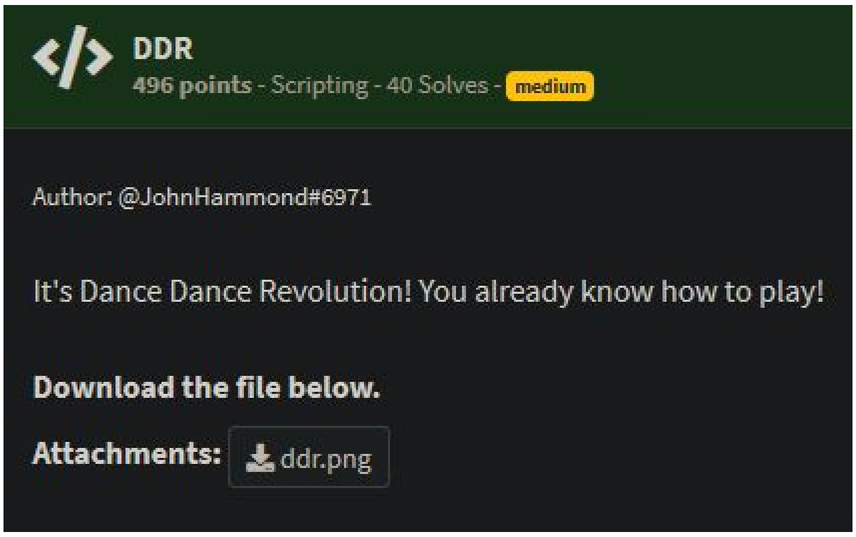
```
    function b() {
      var t;
      module26.default(this, b);
      (t = v.call(this, ...args)).state = {
        output: 'Insert the pin to get the flag',
        text: '',
      };
      t.partKey = 'pgJ2K9PMJFHqzMnqEgL';
      t.cipher64 = 'AA9VAhkGBwNWDQcCBwMJB1ZWVlZRVAENW1RSAwAEAVsDVlIAV00=';

      t.onChangeText = function (n) {
        t.setState({
          text: n,
        });
      };

      t.onPress = function () {
        var n = module401.Base64.toUint8Array(t.cipher64),
          o = module402.sha256.create();
        o.update(t.partKey);
        o.update(t.state.text);

        for (var l = o.hex(), u = '', c = 0; c < n.length; c++) u +=
String.fromCharCode(n[c] ^ l.charCodeAt(c));

        t.setState({
          output: u,
        });
      };
```

Since it only using number for input. We can create JavaScript script inside the same folder to bruteforce the flag.

```
module401 = require('./401'),
module402 = require('./402');

partKey = 'pgJ2K9PMJFHqzMnqEgL';
cipher64 = 'AA9VAhkGBwNWDQcCBwMJB1ZWVlZRVAENW1RSAwAEAVsDVlIAV00=';

var n = module401.Base64.toUint8Array(cipher64);
var o = module402.sha256.create();

for(var i=0; i<999999999; i++)
{
    o = module402.sha256.create();
    o.update(partKey)
    o.update(i.toString()) //4784
    for (var l = o.hex(), u = '', c = 0; c < n.length; c++) u +=
String.fromCharCode(n[c] ^ l.charCodeAt(c));
    if(u.startsWith("flag{"))
    {
        console.log("Pin: "+i.toString()+", output: "+u)
//flag{06754e57e02b0c505149cd1055ba5e0b}
```

```
    }
}
```

Category: Scripting - DDR



DDR

496 points - Scripting - 40 Solves - medium

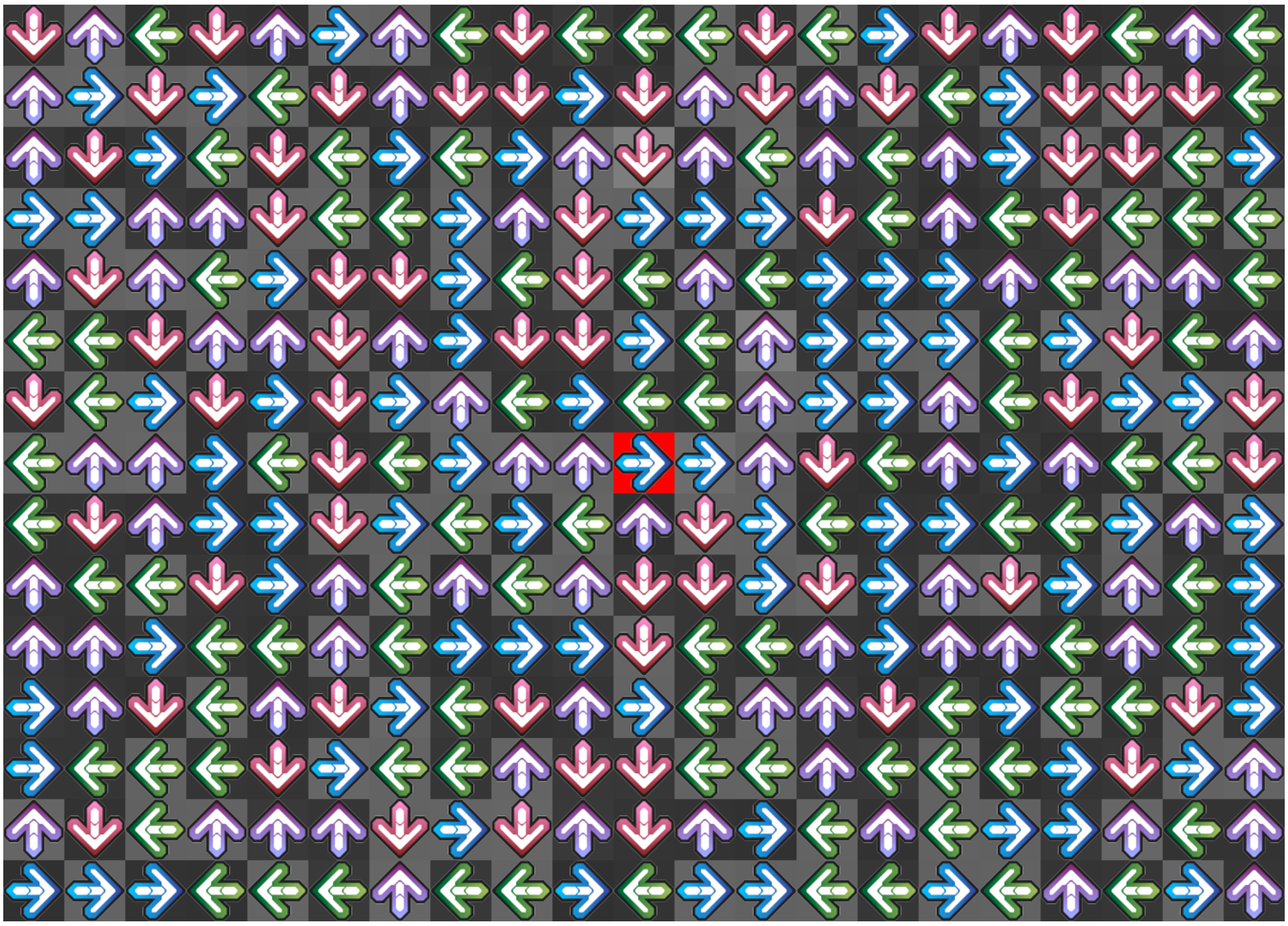
Author: @JohnHammond#6971

It's Dance Dance Revolution! You already know how to play!

Download the file below.

Attachments: ddr.png

Given image below.



Consist of 21x15 box with arrow. Open up in photoshop reveals that all box gray background colour RGB hex value to be `303030` with red, green and blue value to be same and the value are corresponding to ascii for example ascii of `30` hex represent `0`. Therefore script below convert the image into matrix of the ascii character based on the background box colour but only take single colour value.

```
from PIL import Image

def rgb_to_hex(rgb):
    return '%02x' % rgb

im = Image.open("ddr.png")
pix = im.load()
rows, cols = (15, 21)
alphamatrix = [[0 for i in range(cols)] for j in range(rows)]
print(rgb_to_hex((pix[1,1][0])))
print(rgb_to_hex((pix[65,1][0])))
print(rgb_to_hex((pix[129,1][0])))
j = 1
for y in range(15):
    i = 1
    for x in range(21):
        alphamatrix[y][x] = rgb_to_hex((pix[i,j][0]))
        i+=64
    j+=64


for y in range(15):
    for x in range(21):
        if alphamatrix[y][x] == "ff":
            print("@", end="")
        else:
            print(bytes.fromhex(alphamatrix[y][x]).decode('utf-8'),
end="")
    print()
```

After that, we figure it out that the flag start from the middle and follow the arrow to create the flag. Below is the continuation to construct the flag.

```
arrow = "rruuulurrdrrruluuldluulddluullllddrurd"
row = 10
col = 7
print("f", end="")
for x in arrow:
    if x == "l":
        row -= 1
    if x == "r":
        row += 1
    if x == "u":
        col -= 1
    if x == "d":
        col += 1
    print(bytes.fromhex(alphamatrix[col][row]).decode('utf-8'), end="")
```