1. **Mod 26**

* Rot13 là xoay(dịch) mỗi ký tự đi 13 chữ cái trong bảng 26 chữ.
* Dùng web rot13.com để dịch lại xâu đề cho.

1. **Mind your Ps and Qs**

* Tìm hiểu về hệ mã hóa RSA:

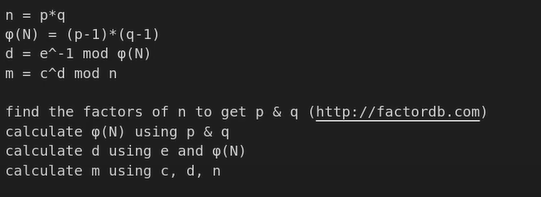
+ RSA là 1 hệ mã hóa bất đối xứng, tức là gồm 2 khóa: public key dùng để mã hóa và private key dùng để giải mã.

+ Trong hệ mã hóa này, bất kỳ ai cũng có thể dùng public key để mã hóa nhưng chỉ người sở hữu private key mới có thể giải mã.

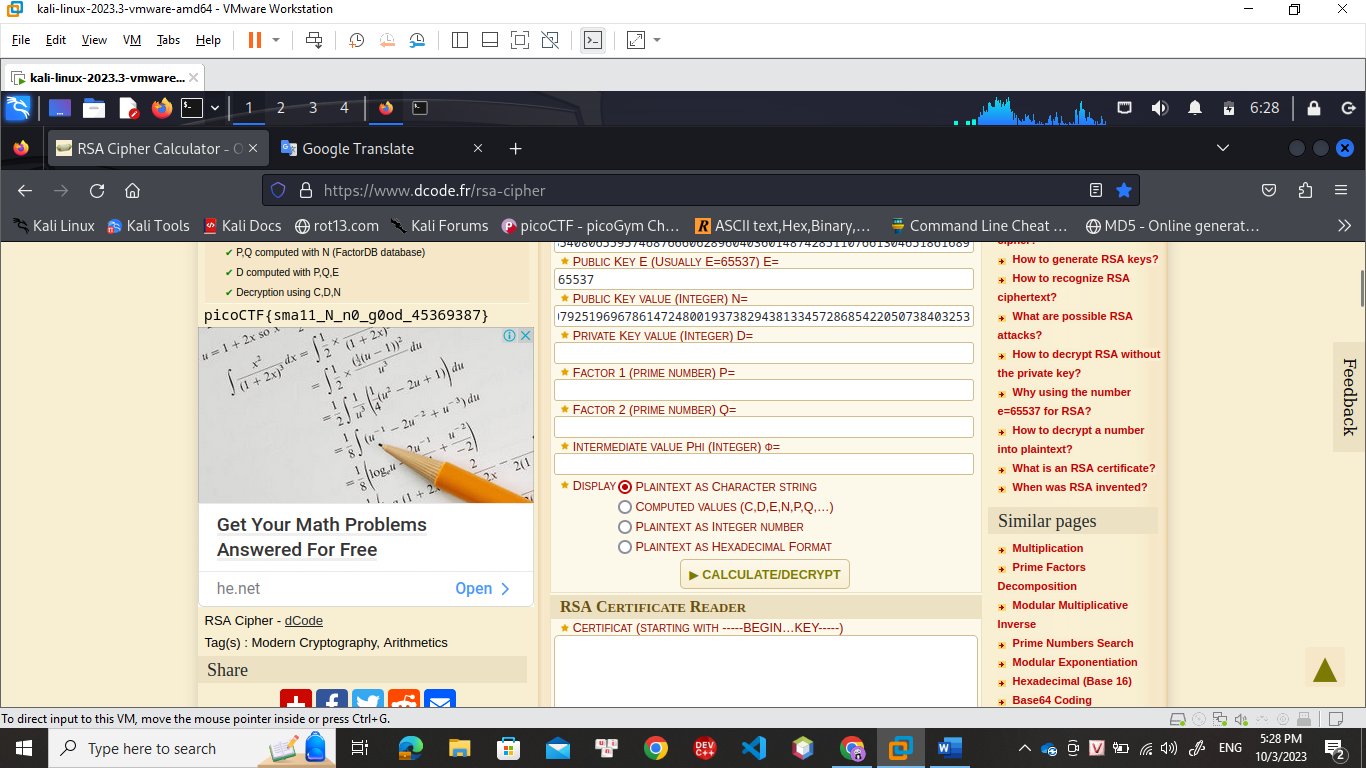
+ Trong hệ mã hóa RSA, public key có thể chia sẻ công khai cho tất cả mọi người, hoạt động của RSA dựa trên 4 bước chính: sinh khóa, chia sẻ key, mã hóa và giải mã.

+ Các ký hiệu trong hệ mã hóa RSA:

* c: ciphertext (văn bản mã hóa)
* n: modulus, là 1 số nguyên lớn được tạo thành bằng cách nhân 2 số nguyên tố lớn p, q lại với nhau.
* e: public exponent (số mũ công khai)
* p, q: 2 số nguyên tố lớn tạo ra n, 2 số này bí mật và không được công khai.
* : hàm euler phi, sử dụng để tính toán khóa công khai và khóa bí mật.
* d: private exponent (số mũ bí mật), sử dụng trong quá trình giải mã ciphertext để khôi phục lại plaintext ban đầu.
* m: plaintext (văn bản gốc)



* Lên google gõ rsa cipher decode, nhập các số đã biết rồi có thể tính các số còn lại hoặc tính luôn plaintext



1. **Easy peasy**

* Tìm hiểu về hệ mã hóa one time pad (OTP)

+ Trên lý thuyết thì one time pad không thể bị phá vỡ.

+ Đây là 1 dạng mã hóa sử dụng key để mã hóa plaintext thành ciphertext.

+ Plaintext XOR key = cipher text

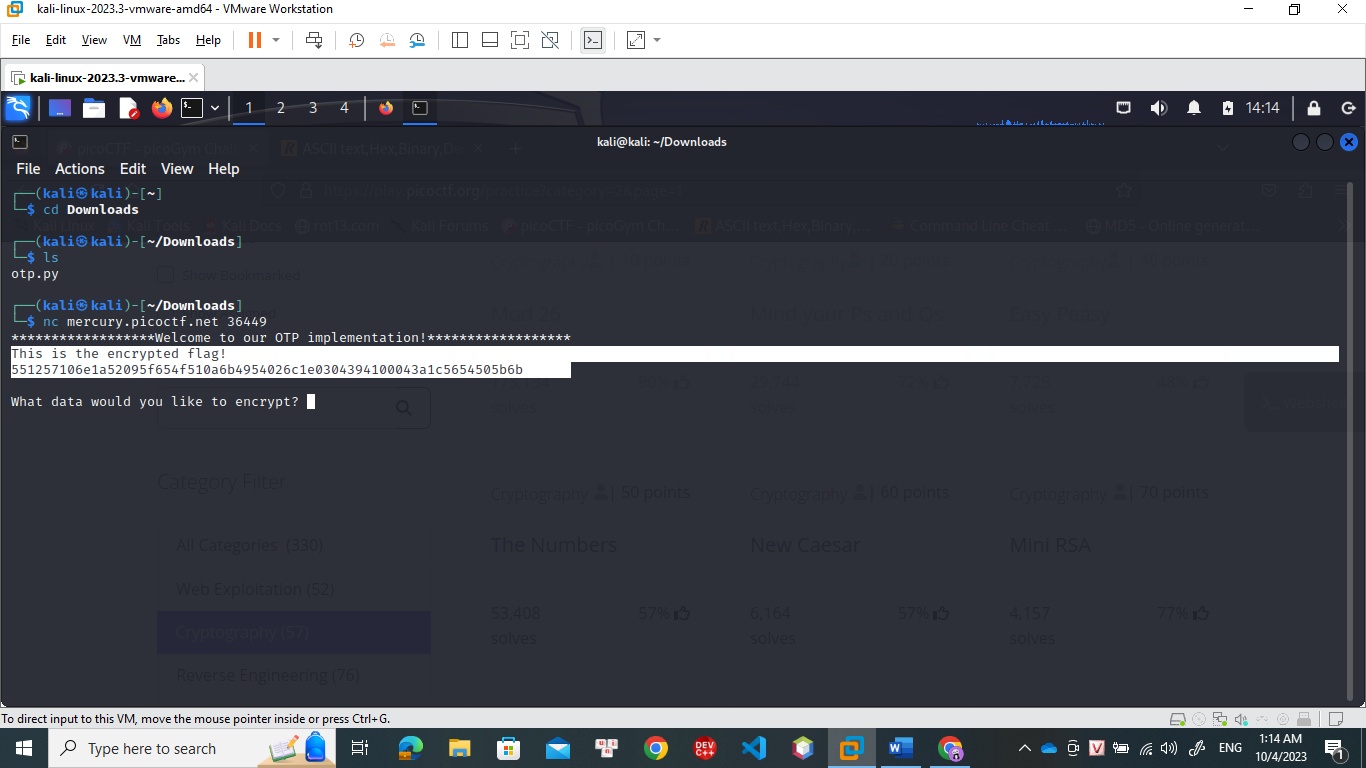
+ Plaintext = key XOR cipher text

+ key phải có độ dài >= độ dài plaintext

* Ở bài này đề đã cho cipher text (sau khi netcat thì thấy cờ đã được mã hóa)
* Nhiệm vụ là phải tìm được key để thực hiện phép xor tìm ra plaintext. Key này phải là key gốc và được tạo từ key trung gian có độ dài bằng với độ dài ciphertext.
* Ta thấy rằng ciphertext có 64 ký tự hexa => độ dài 32 byte vì mỗi 1 byte trong hệ hexa tương ứng với 2 ký tự.
* Mà trong code đề cho thì KEY\_LEN = 50000, đã mất 32 byte để mã dùng cho ciphertext, do đó khi tạo key thì phải bắt đầu trong 49968 byte còn lại và độ dài mỗi lần tạo là 32 byte.
* Sau khi tạo xong key trung gian từ 1 ký tự bất kỳ, phải lấy key đó xor với 32 byte ký tự đó theo mã hexa, kết quả là key gốc.
* Đem key gốc xor với cipher text sẽ ra được plain text cần tìm.

Chi tiết:

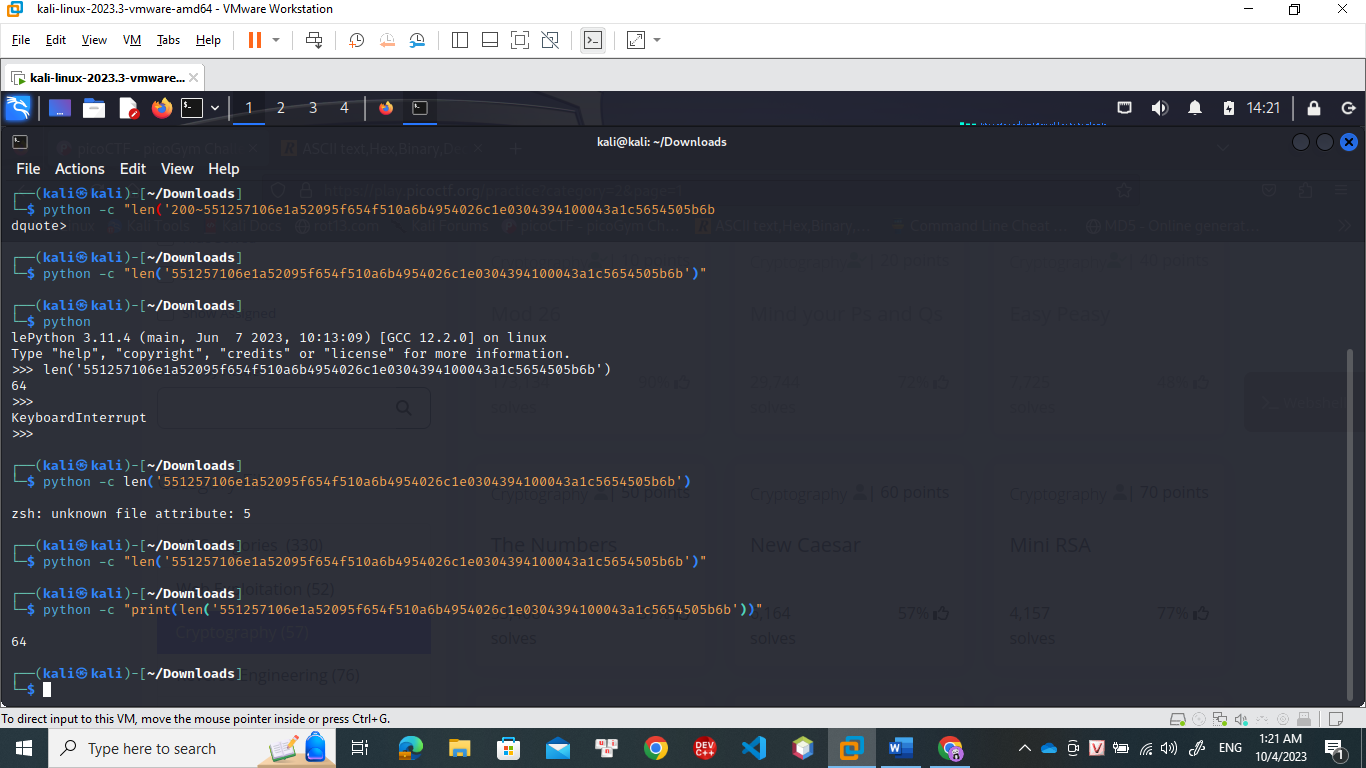
Ciphertext



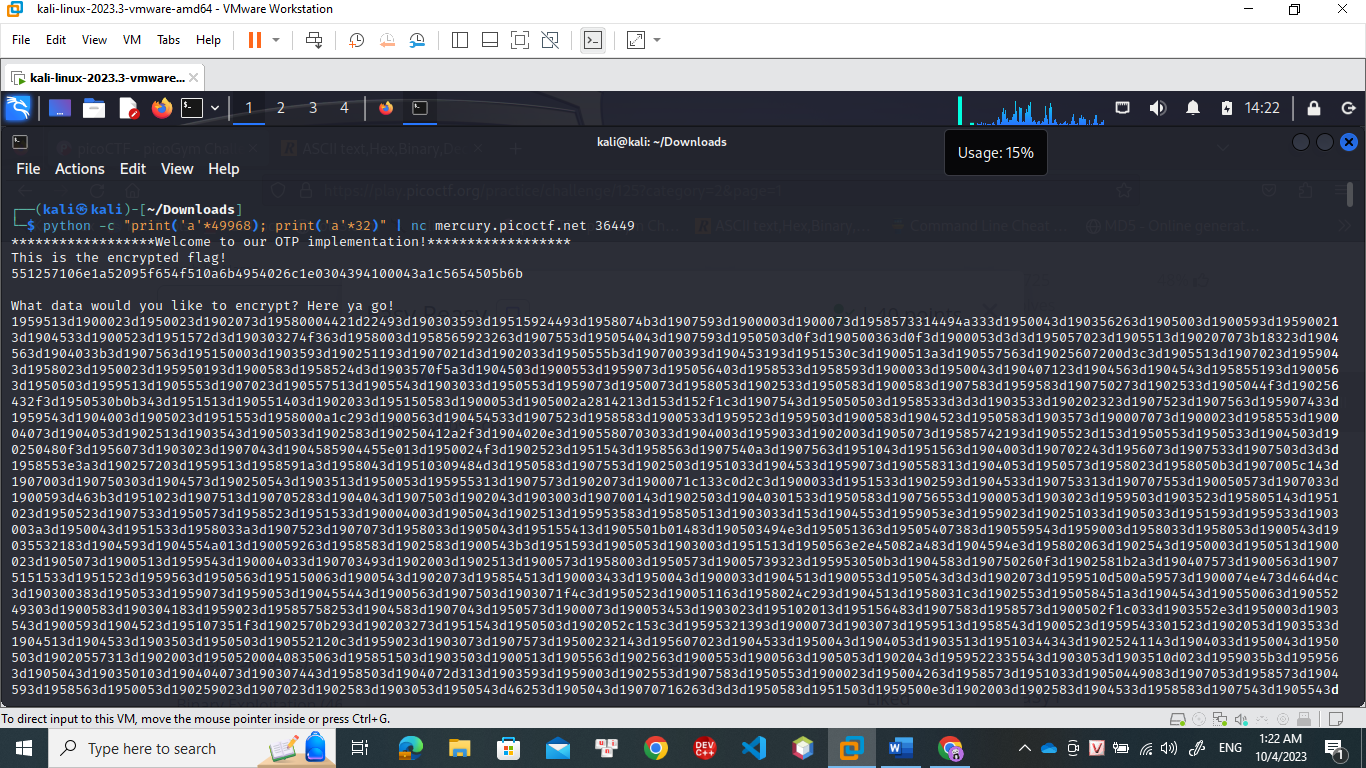
Số ký tự trong ciphertext

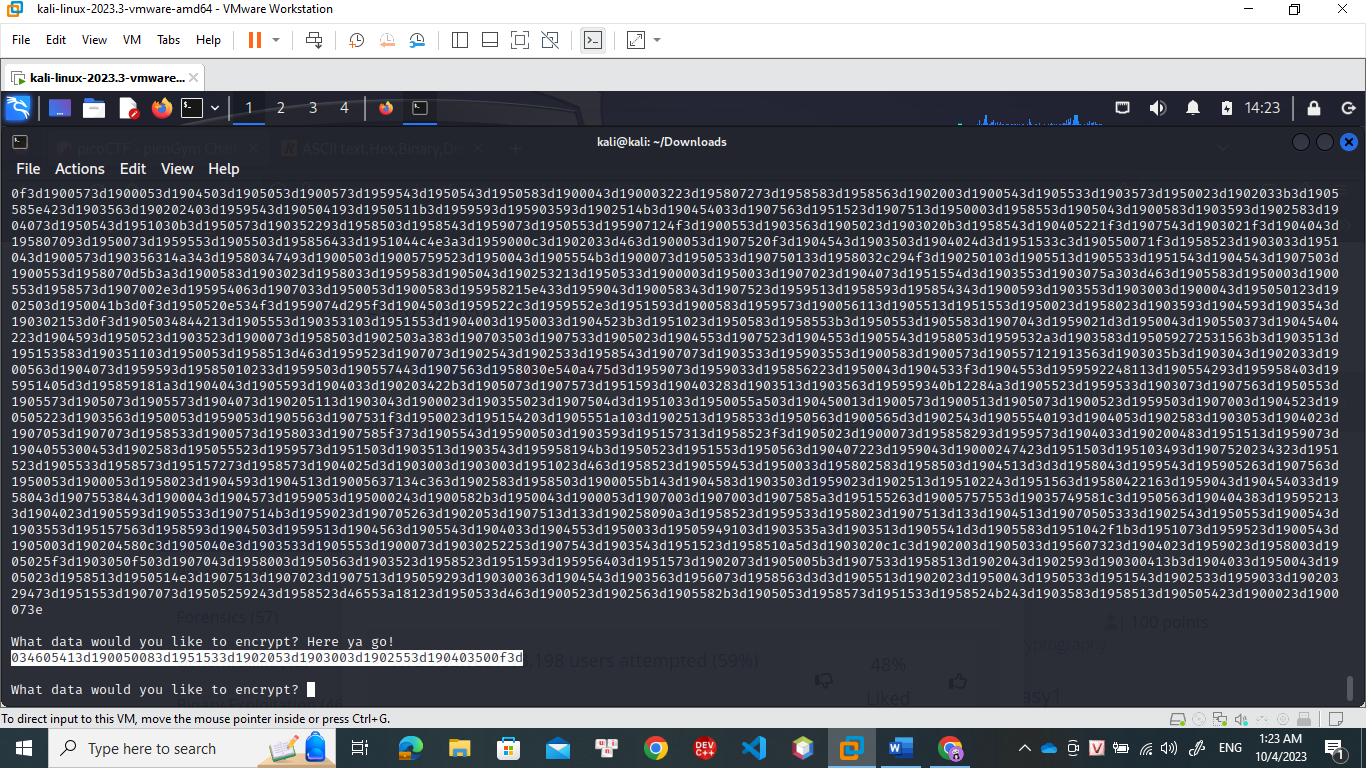
 Có thể kiểm tra bằng lệnh

*python -c print(len('551257106e1a52095f654f510a6b4954026c1e0304394100043a1c5654505b6b'))"*

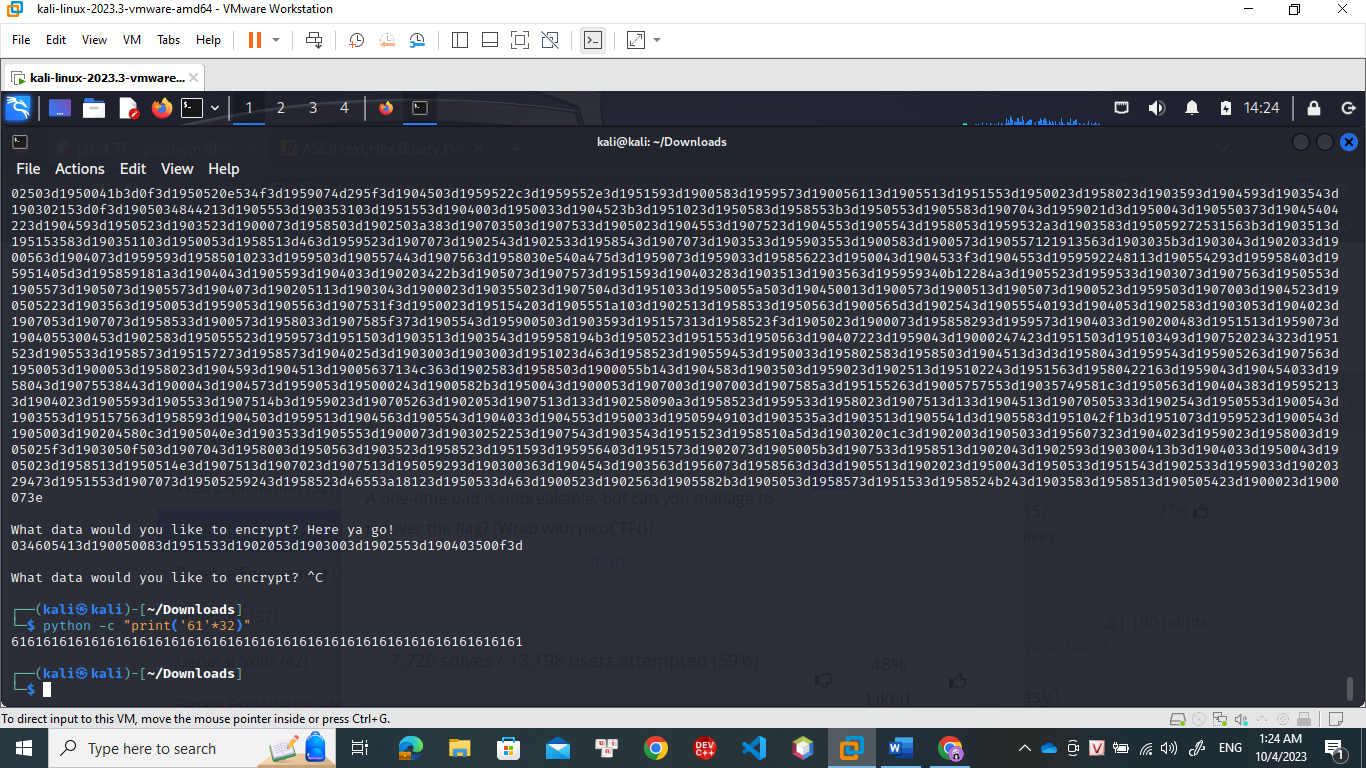


Tạo key trung gian (cipher text của 1 ký tự bất kỳ)





Do sử dụng ký tự a tạo key trung gian nên plaintext của ký tự này là một chuỗi hexa gồm 32 số 61 (vì trong bảng ascii thì a có mã hexa là 61)



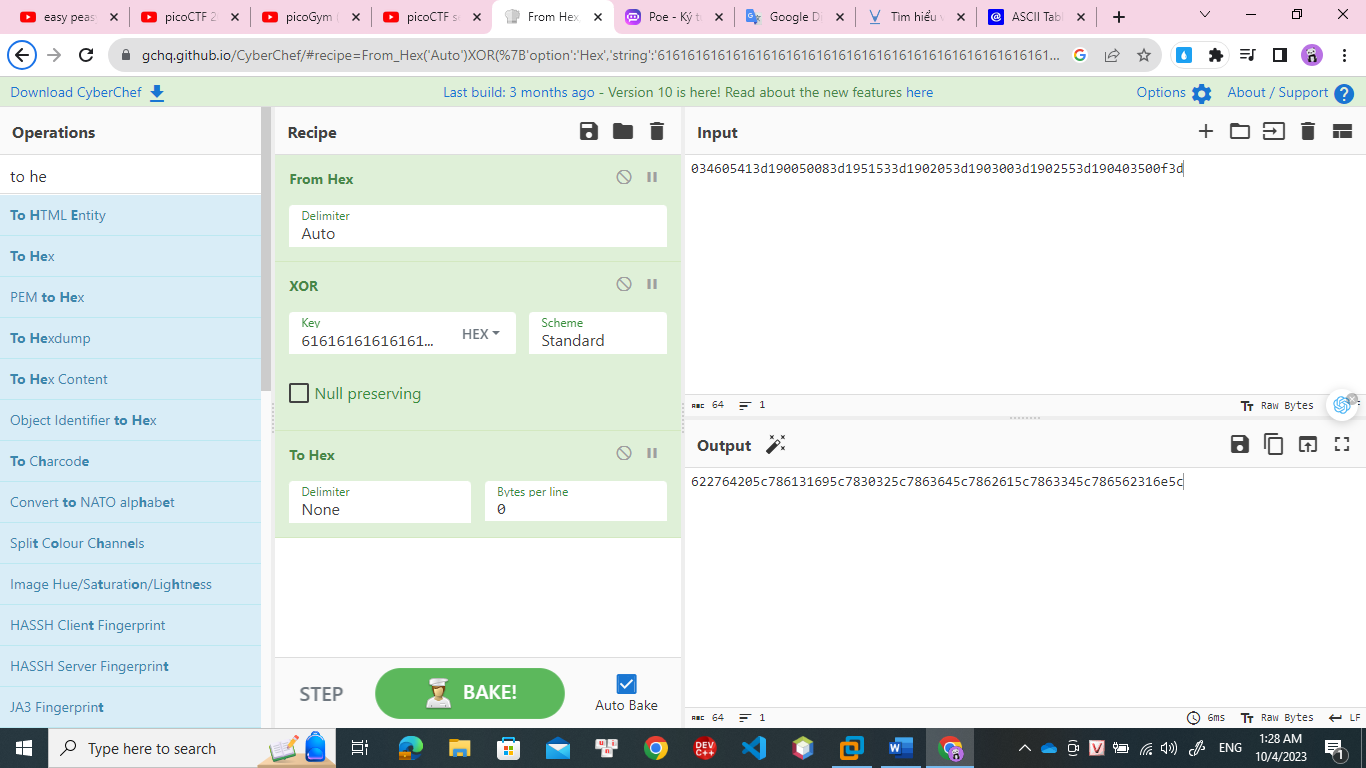
Đem xor plaintext của a với ciphertext của a để tìm key gốc.

Sử dụng cyberchef

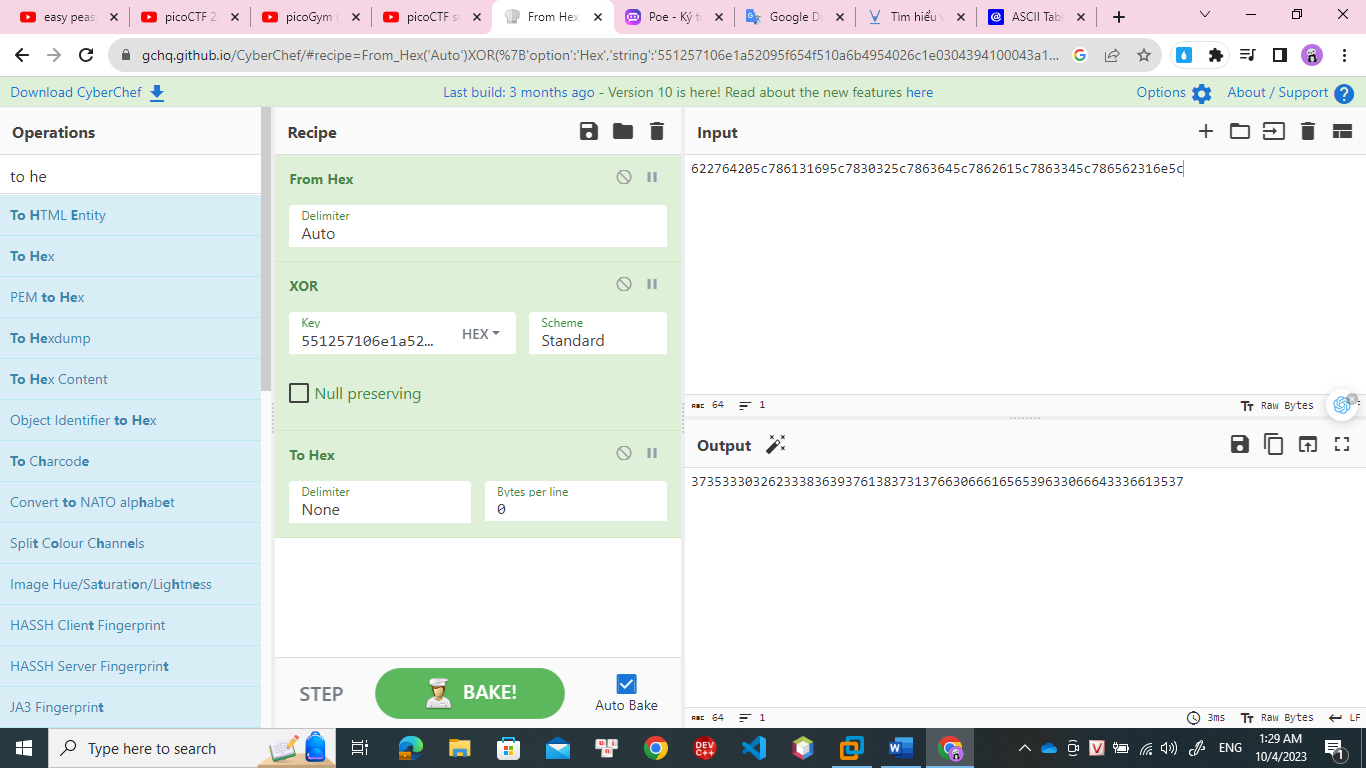
From hex là ciphertext của a

Xor với key là plaintext

Kết quả (key gốc) là to hex



Lấy key gốc xor với ciphertext của flag



Kết quả đang ở dạng hex, cần chuyển sang mã ascii submit.

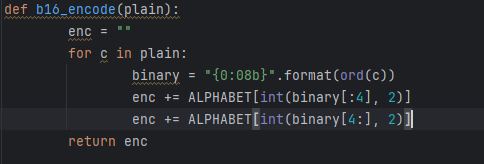
1. **New Caesar:**

* Bài này cờ (plaintext) được mã hóa như sau:

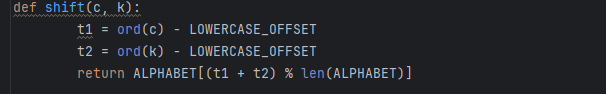
Xét từng ký tự trong plaintext, chuyển ký tự đó sang nhị phân (8 bit), chia đôi cụm nhị phân đó thành 2 phần mỗi phần 4 bit, với mỗi phần thì lại chuyển nó thành chữ (ASCII).

* Giải mã như sau: Xét cụm mã hóa ciphertext (cụm đề bài cho)
* Mỗi ký tự của ciphertext giải mã bằng cách dịch chuyển theo key.
* Key trong bài này là ký tự thuộc từ a-p (16 chữ cái), giả sử dịch chuyển theo a tức là dịch đi 0 ký tự (giữ nguyên), theo b là dịch đi 1 ký tự => Tương tự phép Rot. Lưu ý rằng key này được sinh ngẫu nhiên trong 16 ký tự nên lúc in ra cờ thì phải tìm cờ đọc được để submit.
* Sau khi dịch xong từng ký tự trong ciphertext, ta được chuỗi plaintext đã dịch, tuy nhiên do cờ được mã hóa bằng cách chia đôi bit trong từng ký tự nên cần thêm hàm giải mã điều này.
* Duyệt cụm plaintext đã dịch với bước nhảy 2 (trong mỗi lần duyệt xét 2 ký tự). Tìm giá trị số nguyên của 2 ký tự rồi lấy t1 dịch trái 4 bit (thêm 4 bit 0 ở cuối) cộng với t2, mục đích là để tạo thành chuỗi 8 bit. Sau đó chuyển nó sang mã ASCII.
* Chi tiết

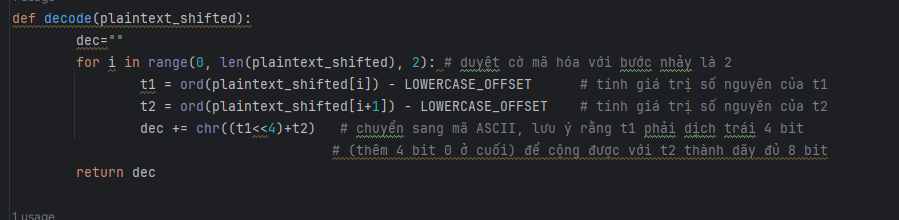
Hàm mã hóa:



Hàm dịch theo key:



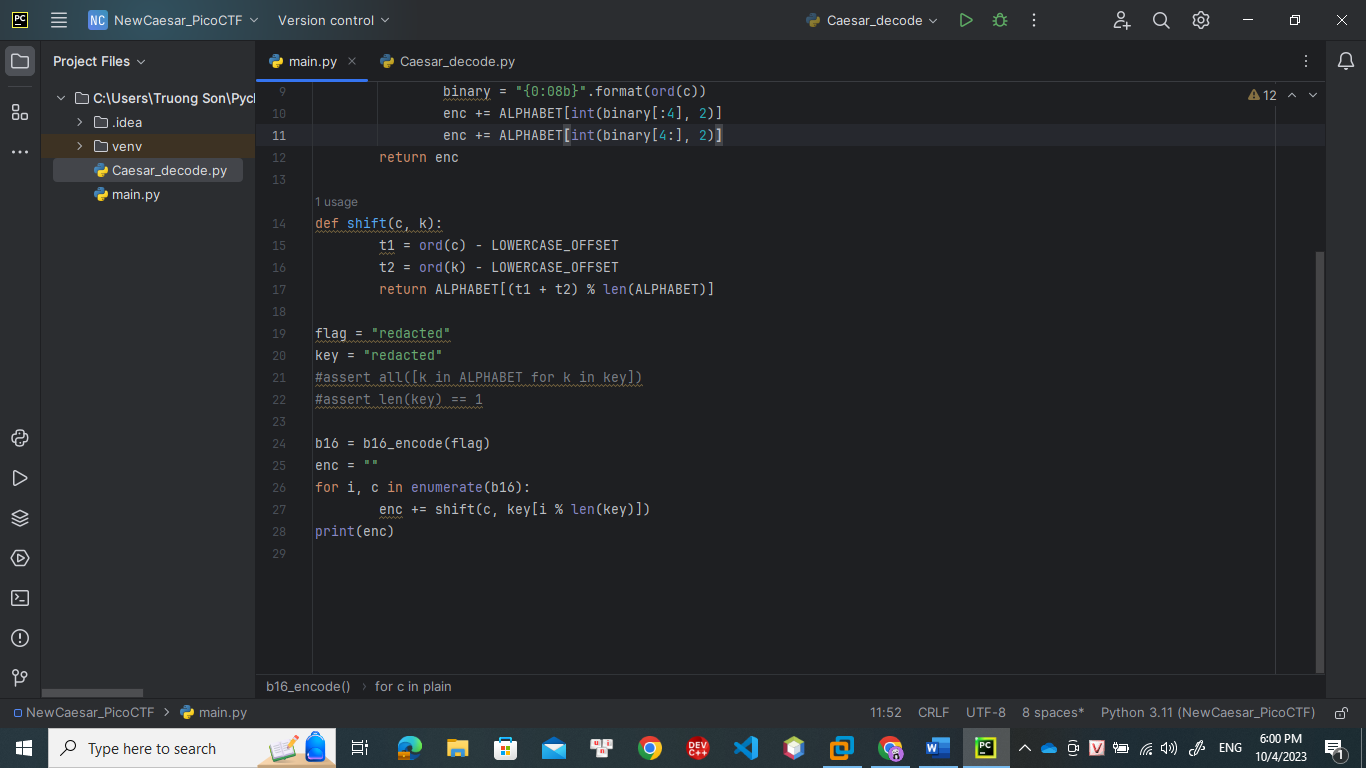
Hàm giải mã:



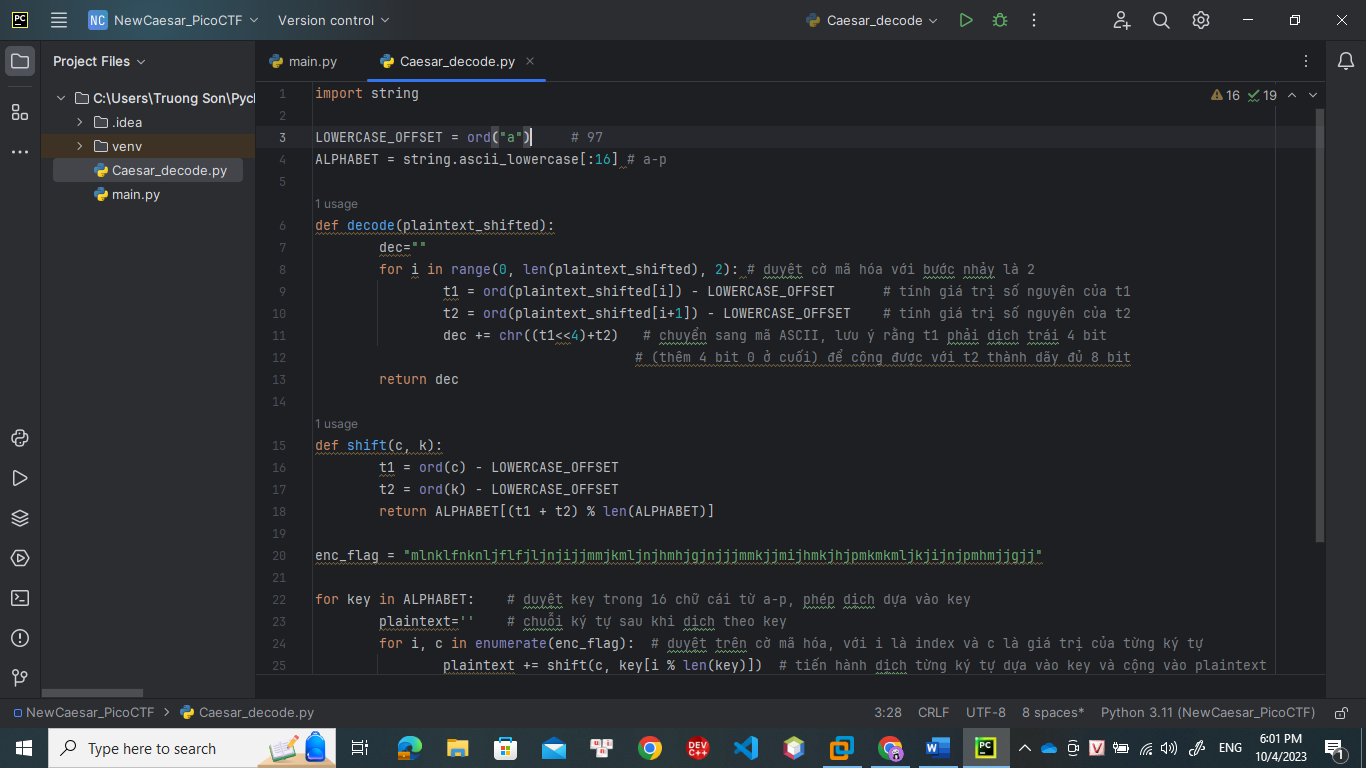
Full code:

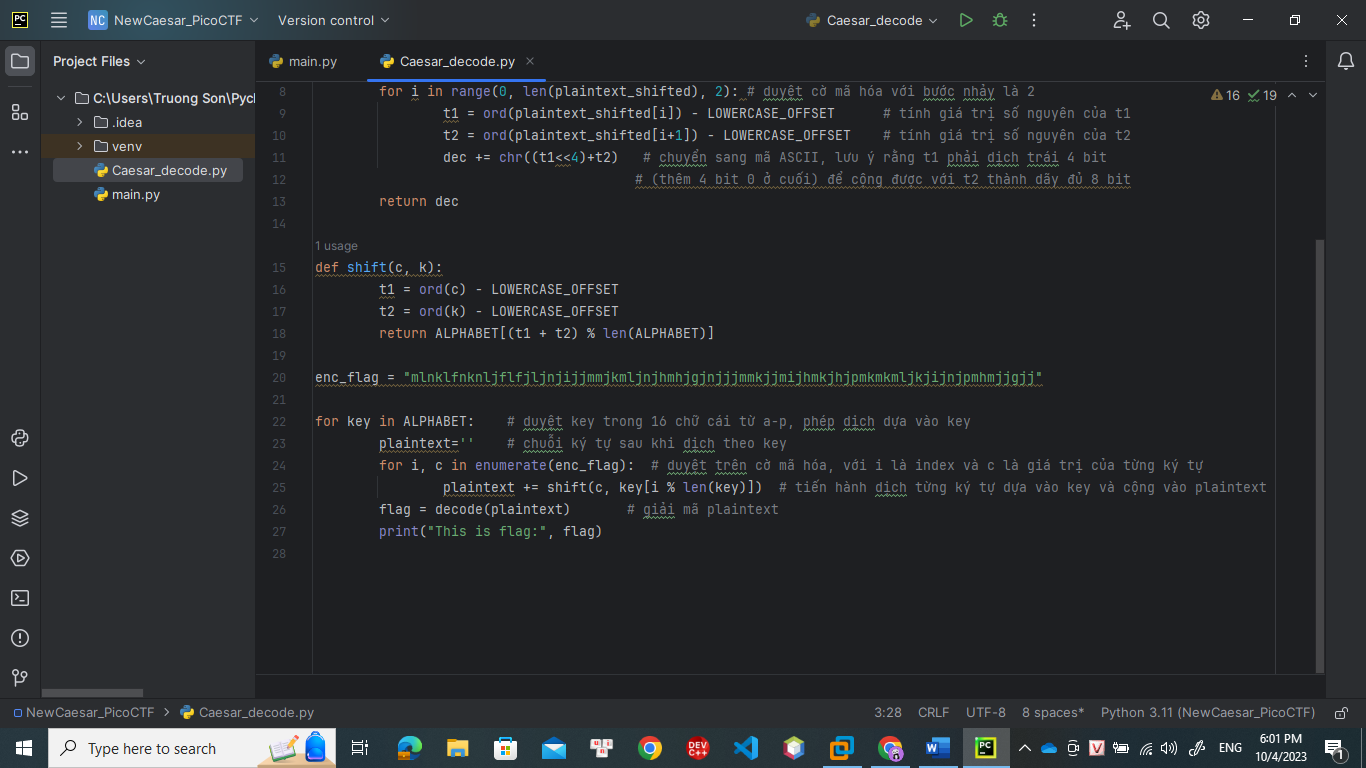
Code đề cho





Code giải mã





1. **The Numbers**

Chuyển số sang chữ (bảng chữ cái tiếng Anh)

1. **Easy1**

Table cho ta biết đây là Vigenere Cipher

Vigenere Cipher (Dạng bảng): Sẽ cho 1 ciphertext, 1 key và 1 table. Cách giải mã là dùng table để ánh xạ từng chữ cái. Với từng ký tự ciphertext ta nhìn vào hàng của ký tự đó, còn với từng ký tự cua key ta nhìn vào cột, giao nhau tại đâu thì đó chính là ký tự được giải mã.

Có thể dùng web Vigenera Cipher decode để giải mã.

1. **13**

Rot13

1. **caesar**

Lên web caesar decode để giải mã bằng cách brute force tất cả các trường hợp của ciphertext (tìm cụm từ có nghĩa rồi submit)

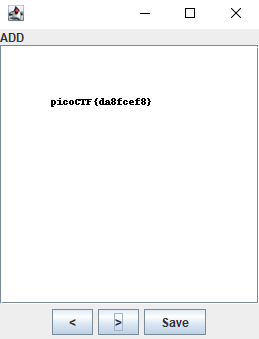
1. **Pixelated**

Kỹ thuật mã hóa dùng trong bài này là tách plaintext ra rồi đặt trong 2 ảnh khác nhau.

Do đó để giải mã thì cần xếp chồng 2 ảnh lên nhau.

Dùng Stegsolve để giải quyết bài này.

Mở stegsolve -> File -> Open -> chọn ảnh 1 -> Analyze -> image combiner -> chọn ảnh 2

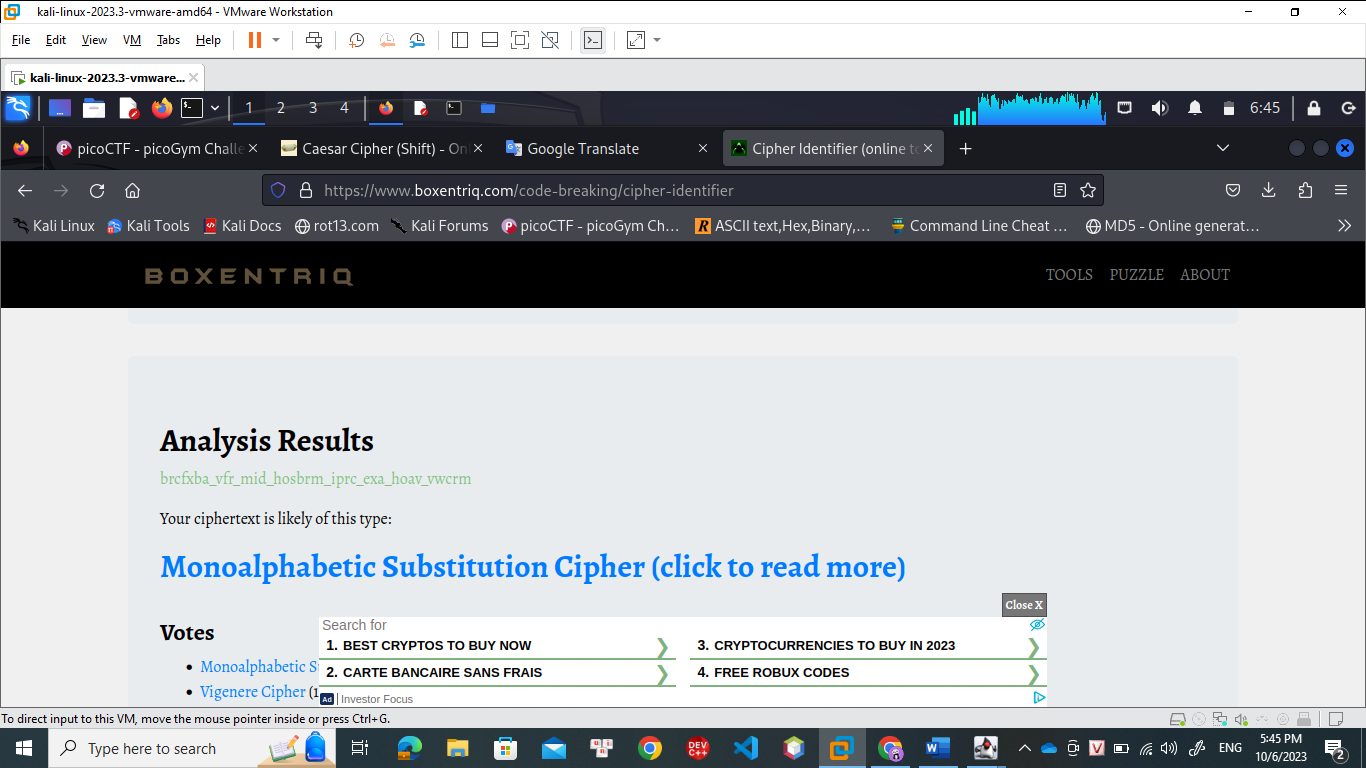


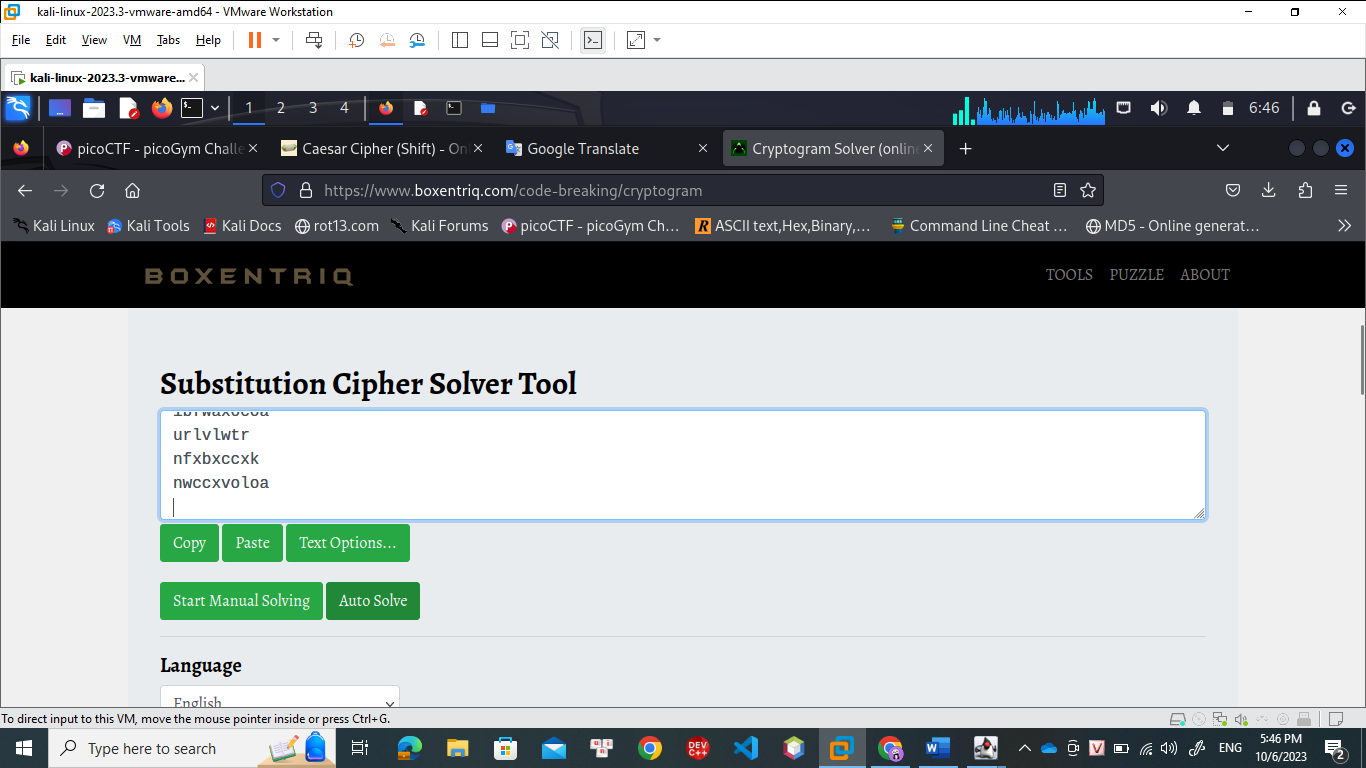
1. **spelling-quiz**

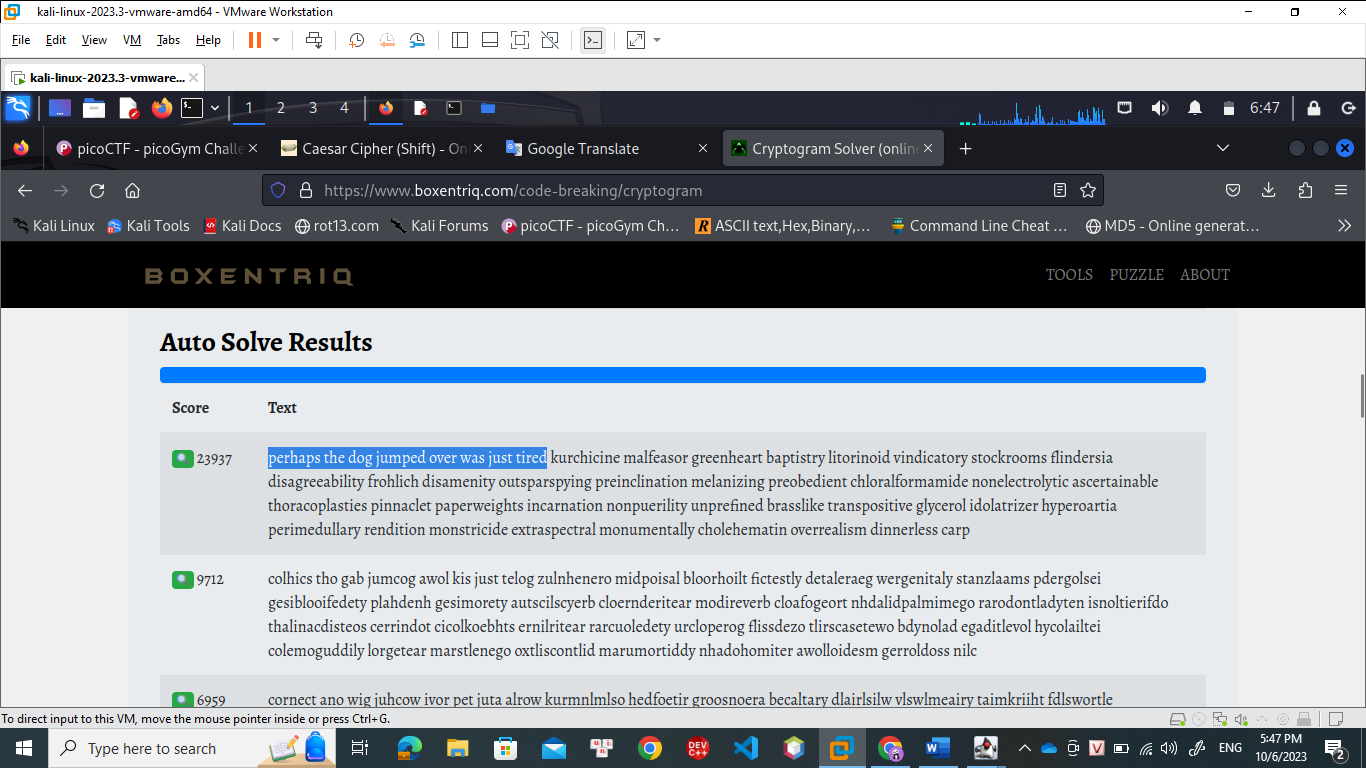
Kỹ thuật mã hóa dùng trong bài này là dịch chuyển ký tự, khá giống mật mã caesar nhưng bài này có key để giải mã.

Lên google gõ cipher decode rồi vào Boxentriq. com, vào Cipher Indentifier tools cop ciphertext vào để nhận diện là kiểu mã hóa nào. => monoalpha…

Cop cả ciphertext và studyguide vào rồi autosolve, cụm từ có khả năng là plaintext sẽ xuất hiện ở đầu.

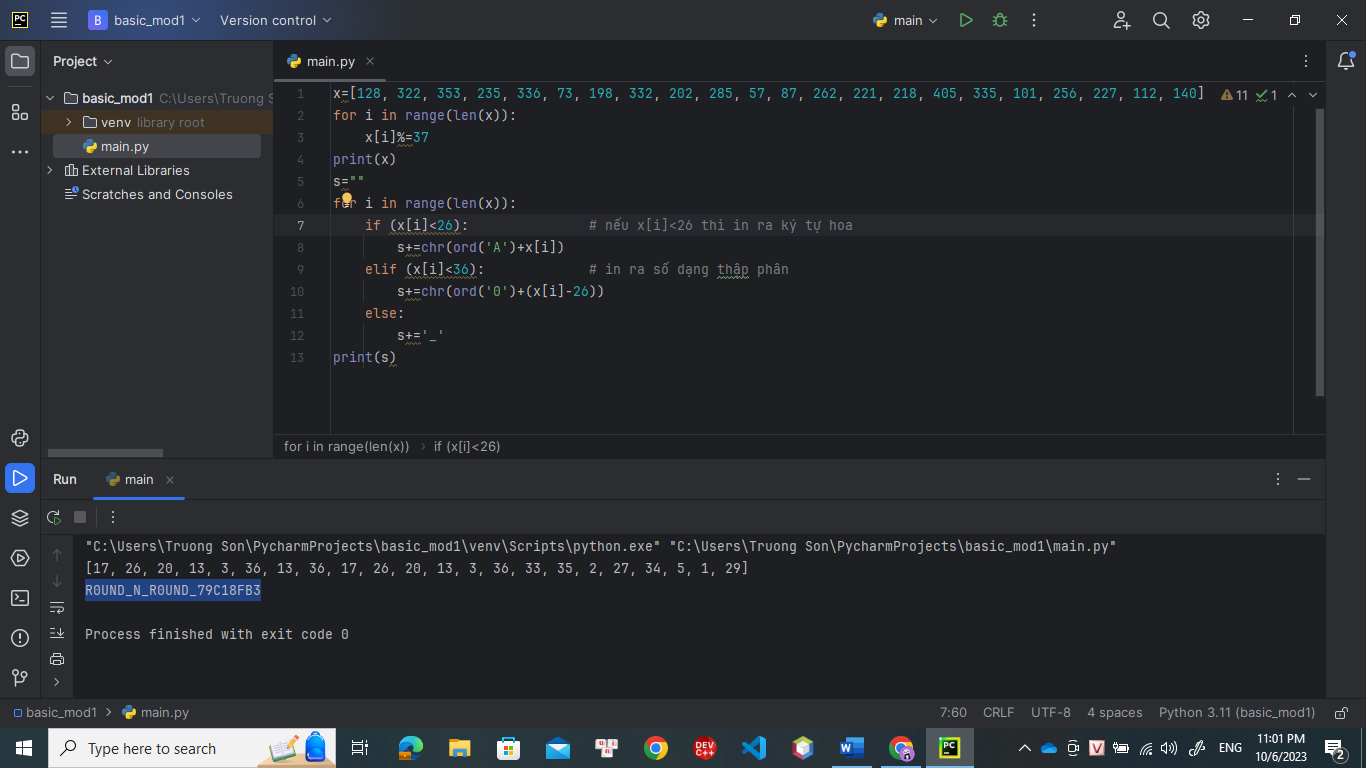






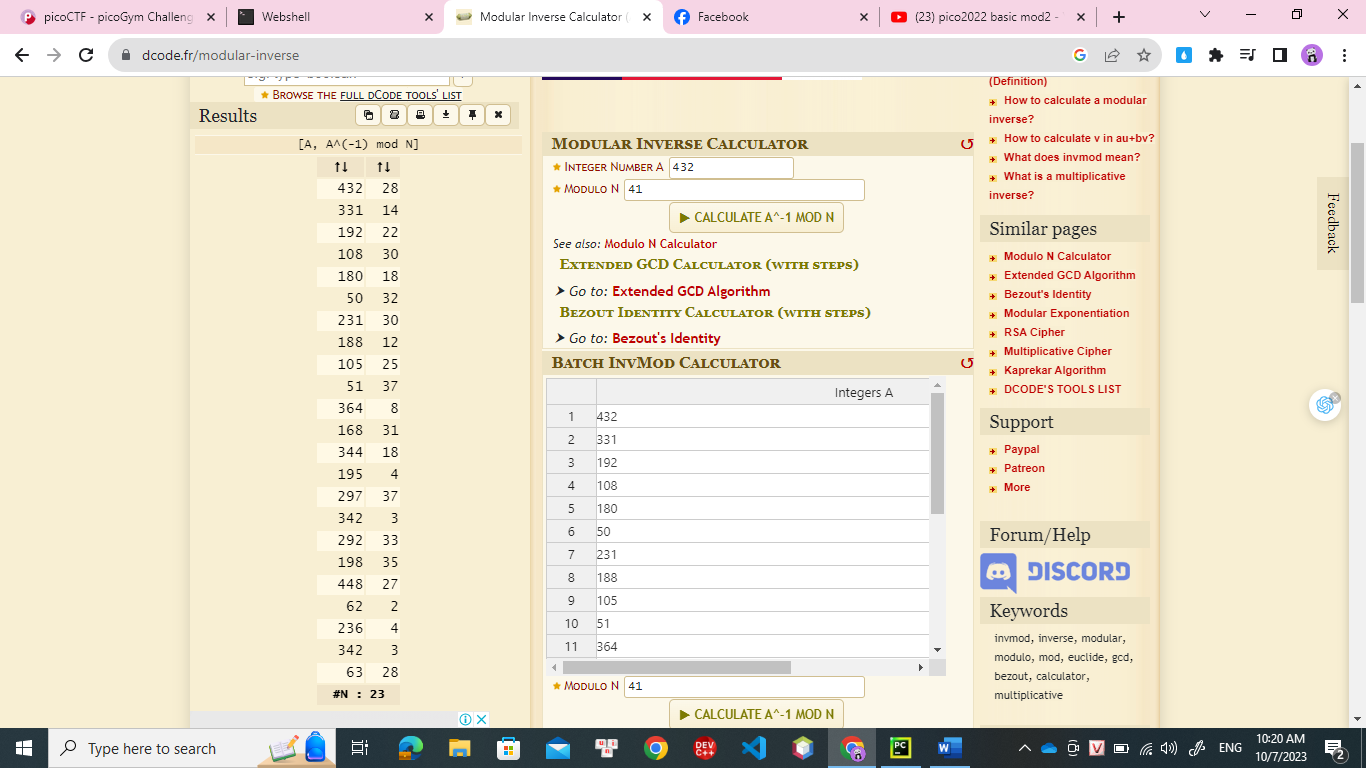
Thêm dấu \_ giữa các từ rồi cho vào picoCTF{}

1. **basic-mod1**

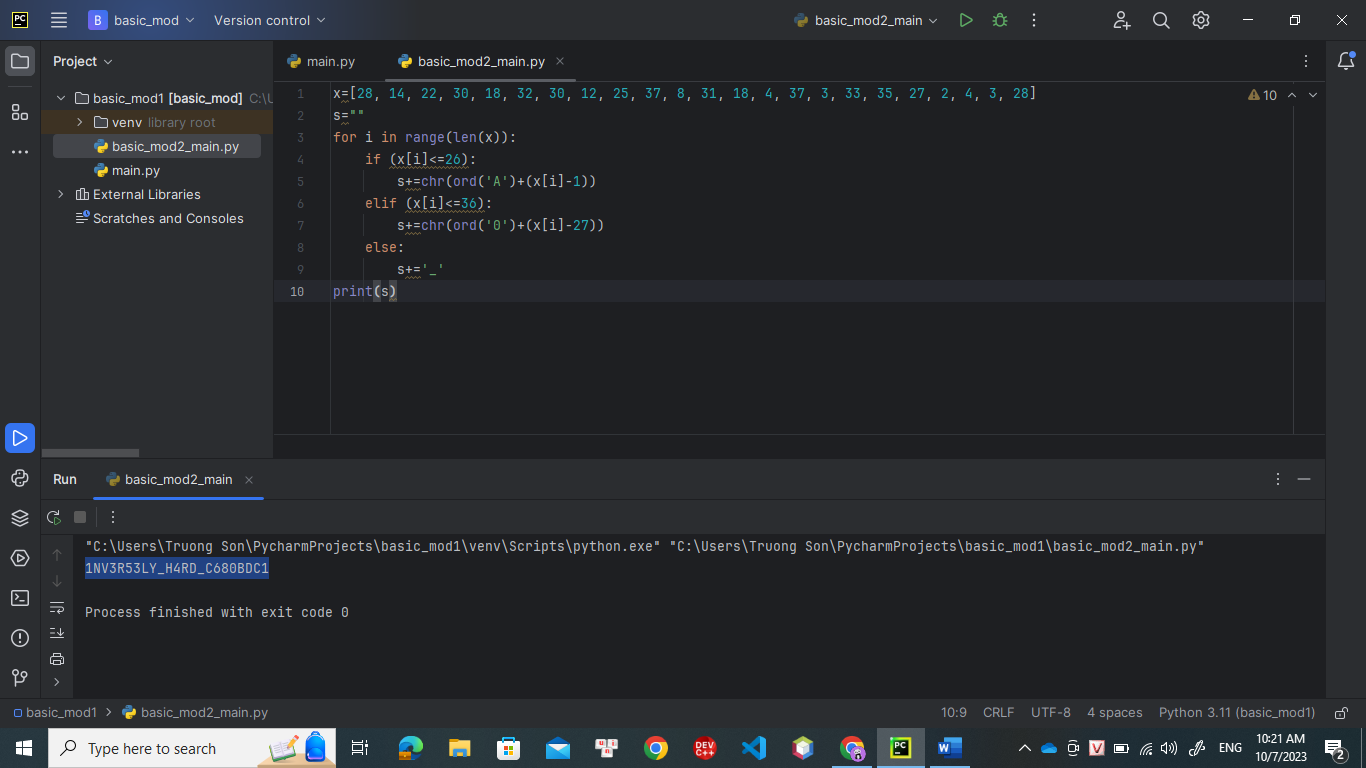


1. **basic-mod2**

Cop đoạn mã đề cho rồi decode theo modular 41 inverse



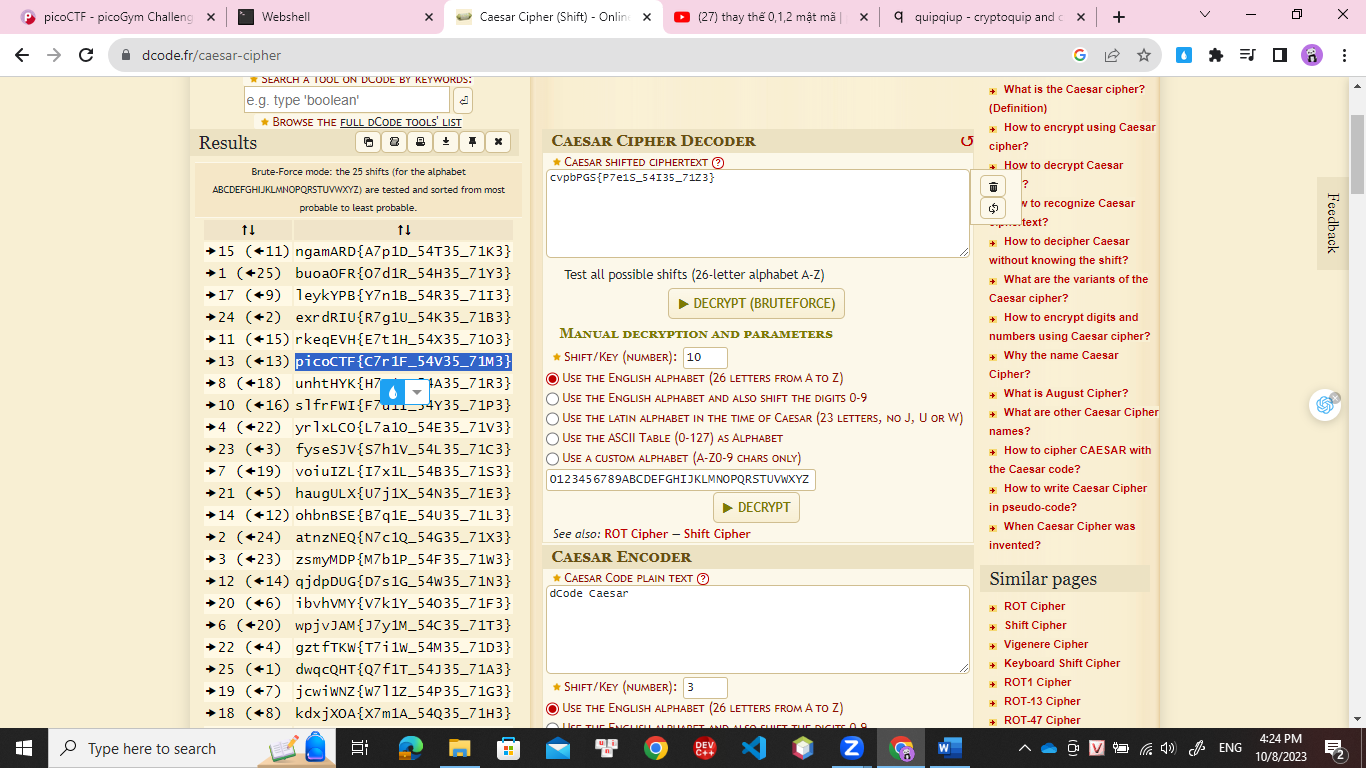
Python giải quyết yêu cầu đề bài



Chú ý ở bài trước là 0-25 thì in ra ký tự, còn ở bài này là 1-26, do đó phải lấy ord(‘A’)+(x[i]-1), trường hợp in ra số cũng tương tự.

1. **credstuff:**

* Giải nén file tải về, thấy có 2 file txt là user và password.
* Mỗi user tương ứng với 1 password.
* Tìm pass ứng với user cultiris.
* Thử decode theo caesar cipher thì thấy cờ pico xuất hiện



1. **Morse code:**

Chuyển mã morse dạng audio sang text.

1. **Rail fence:**

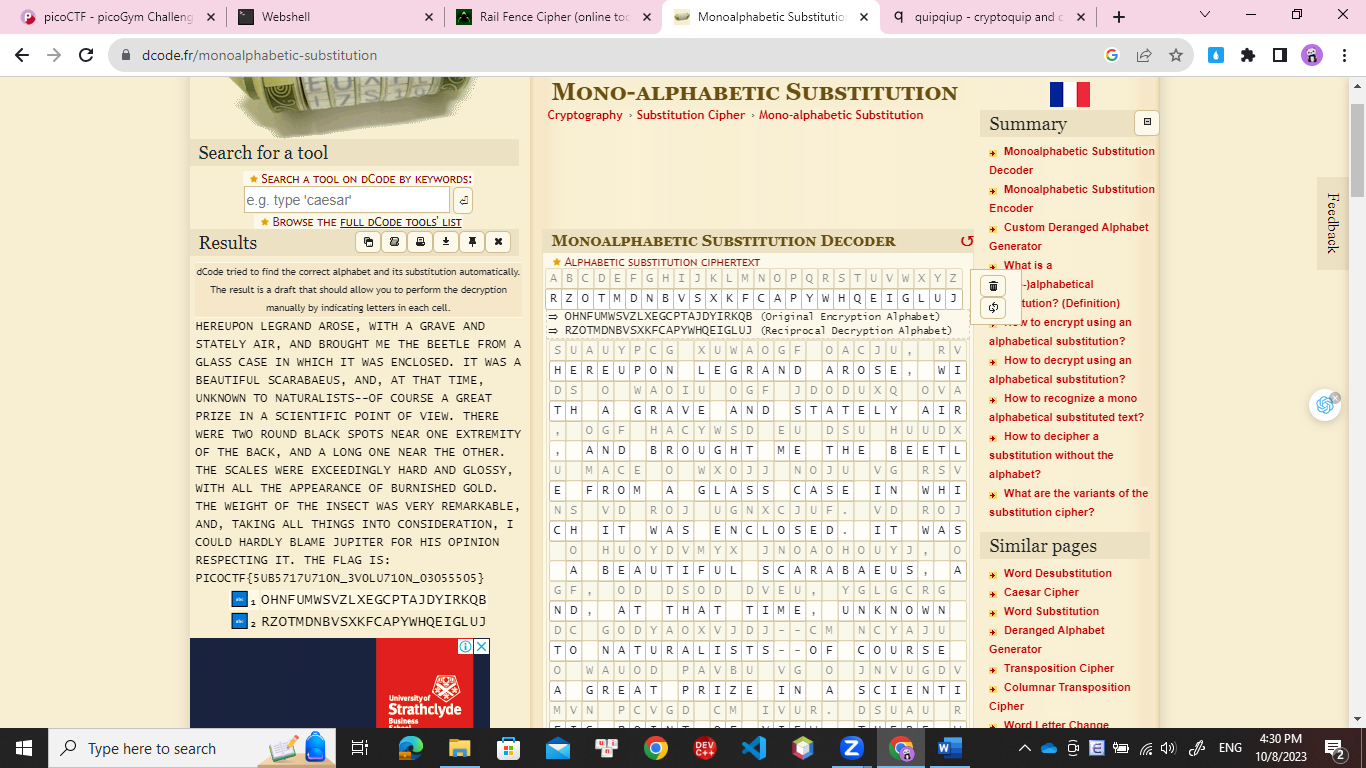
Mật mã rail fence (mã hóa theo kiểu hàng rào)

Giải mã cipher text với 4 rail

1. **Substitution 0**

Gõ cipher analyzer để xem đây là loại thay thế nào thì thấy là mono…

Gõ mono… decode, cop ciphertext và key vào rồi decode



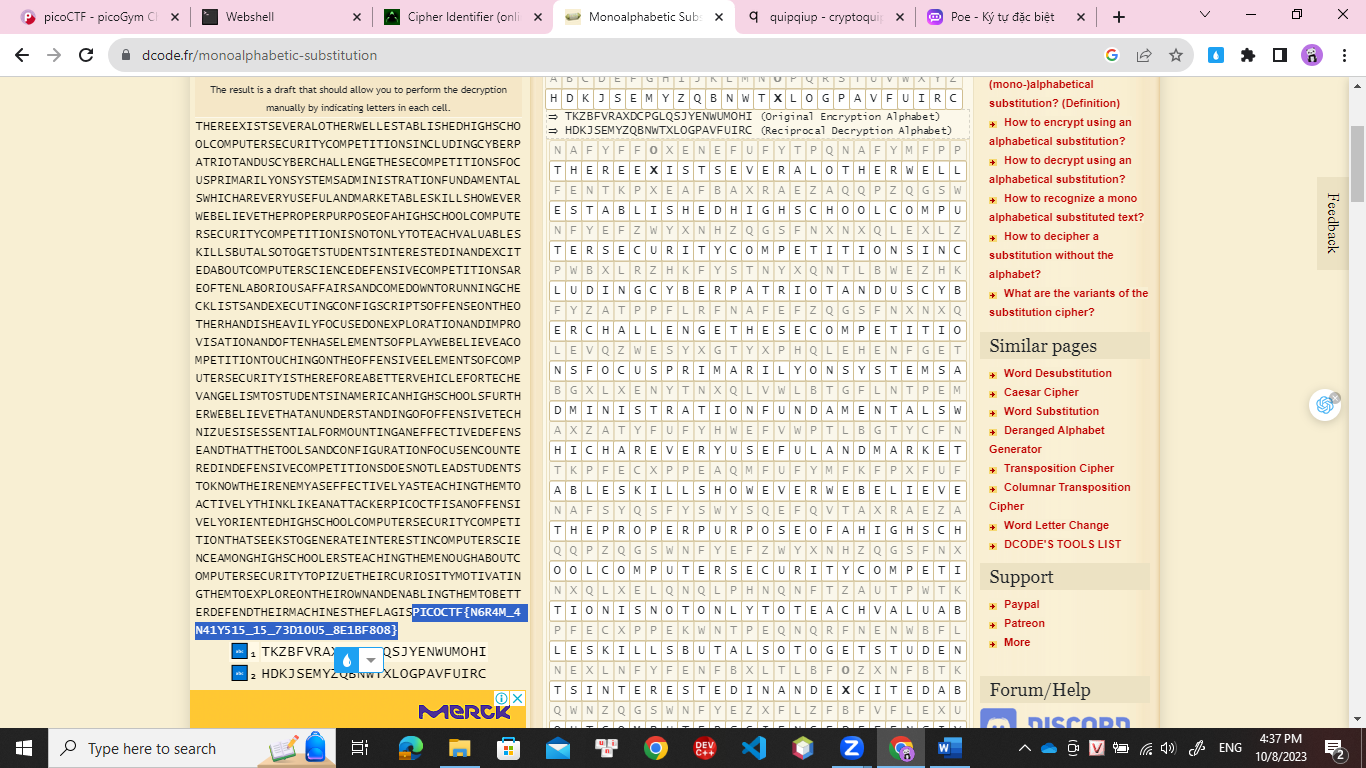
1. **Substitution 1:**

Quipquip: công cụ giải mã các mật mã chuyển vị thông qua phân tích tần suất xuất hiện của các ký tự trong ciphertext, càng dài thì càng dễ decode.

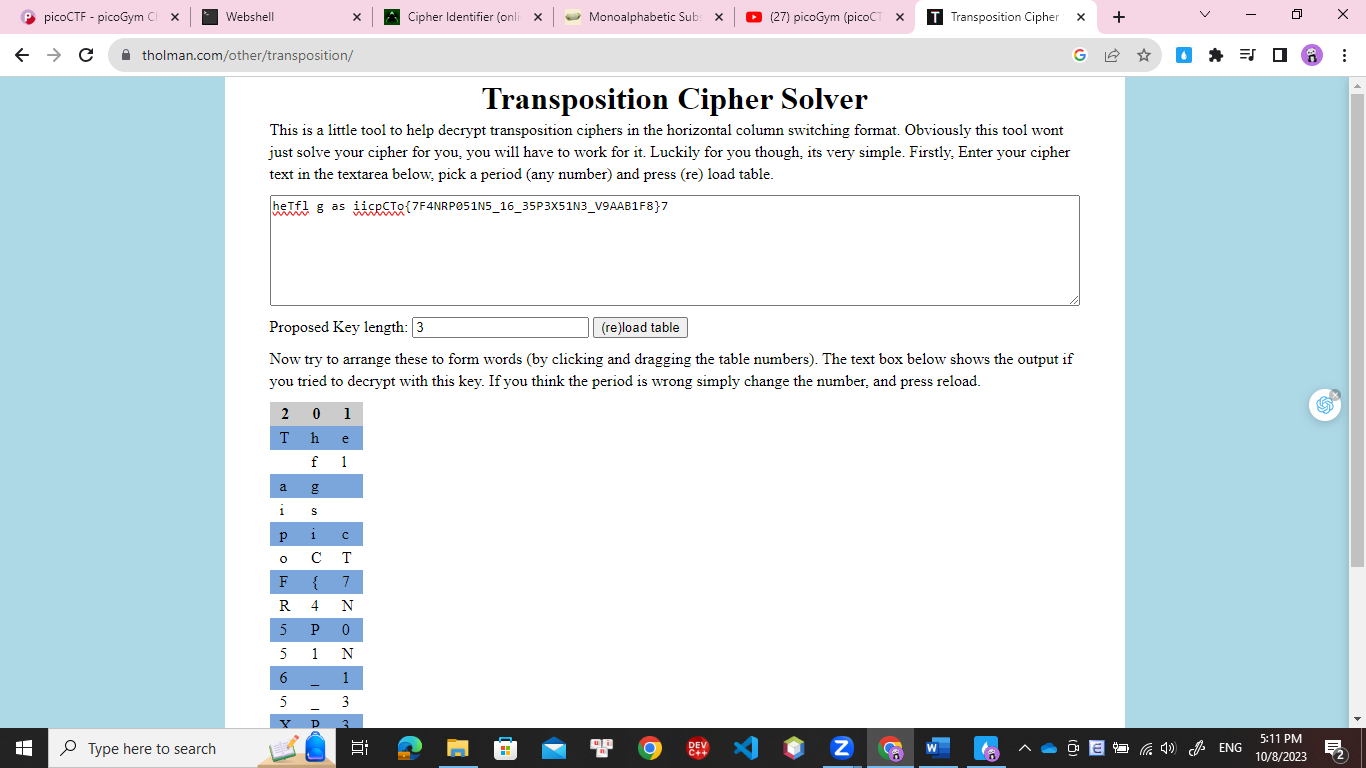
1. **Substitution 2:**

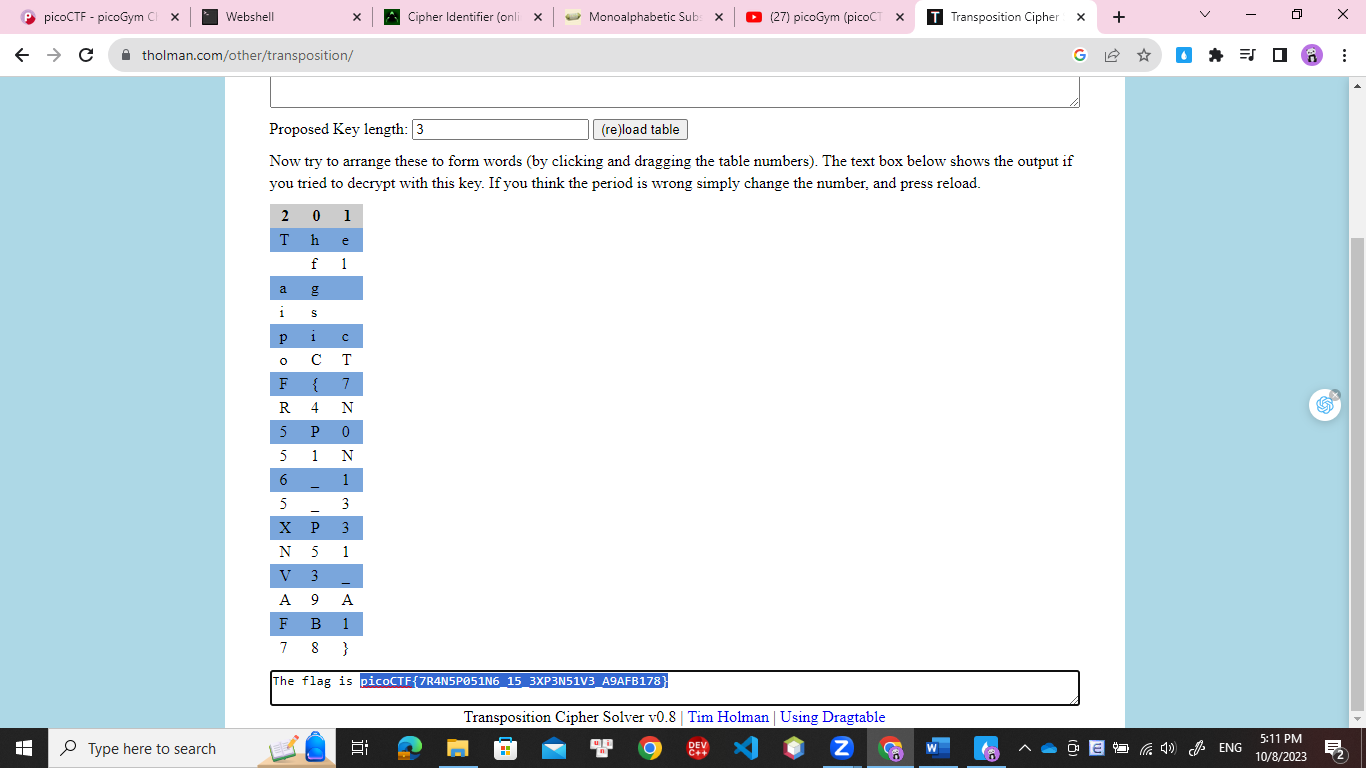
Cipher analyzer để xem là loại mã hóa gì => mono…

Google mono… decode, cop vào rồi decode

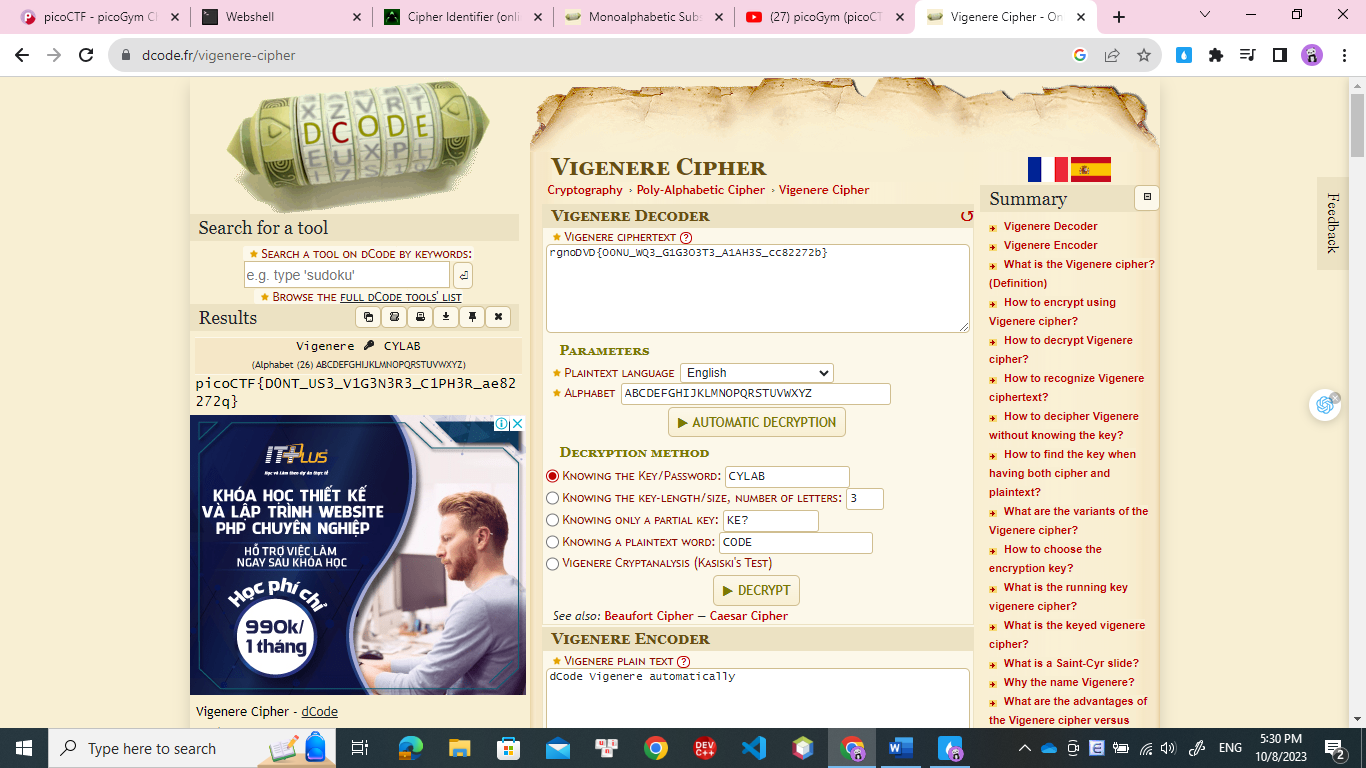


1. **Transposition-trial**





1. **Vigenere cipher:**



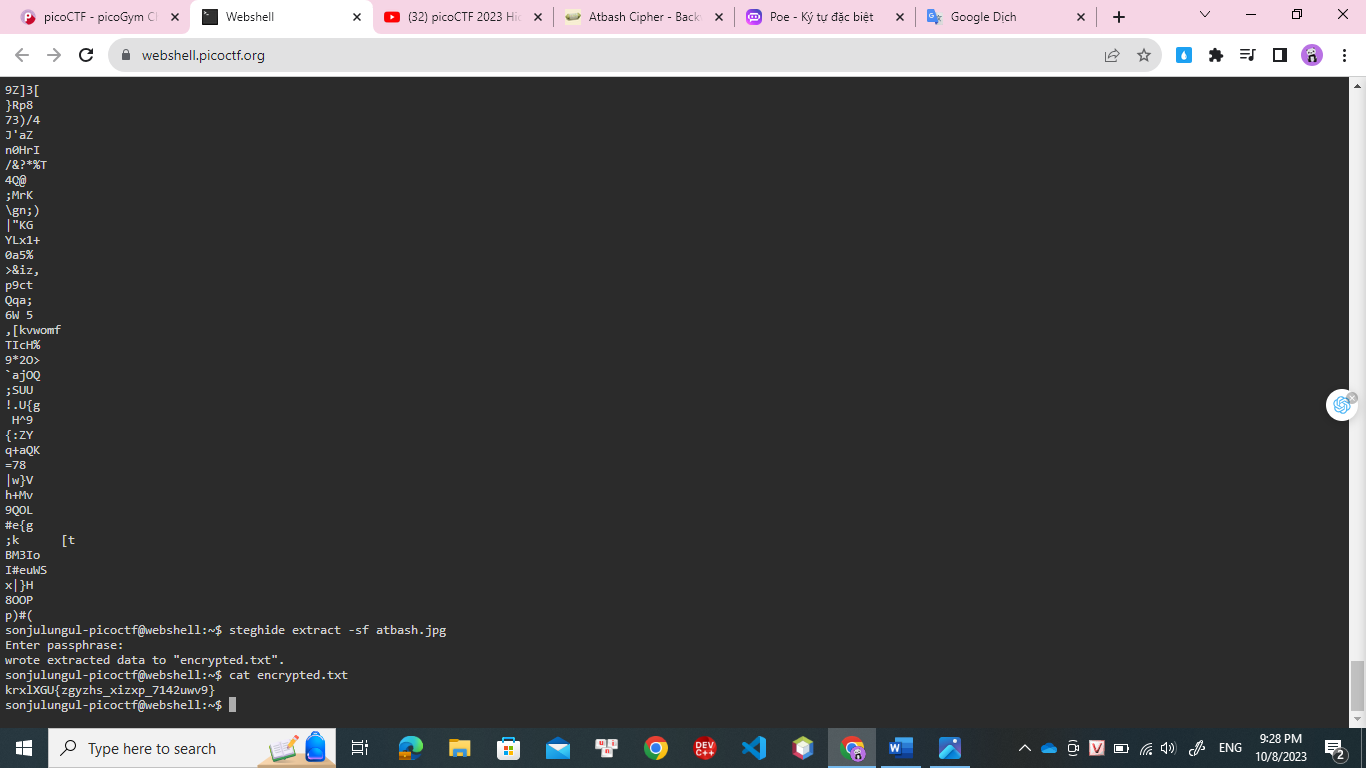
1. **HideToSee**

Atbash cipher: phương pháp mã hóa đơn giản bằng cách dùng các ký tự đối xứng. (a-z, b-y…)

Trong bài này, file tải về định dạng jpg là 1 file ảnh, để trích xuất nội dung 1 file ảnh có 2 cách:

* strings <tên file>
* steghide extract -sf <tên file>

dùng câu lệnh số 2 thì yêu cầu nhập mật khẩu (nếu cần), câu lệnh này sẽ trích xuất các thông tin ẩn trong 1 file ảnh hoặc âm thanh.



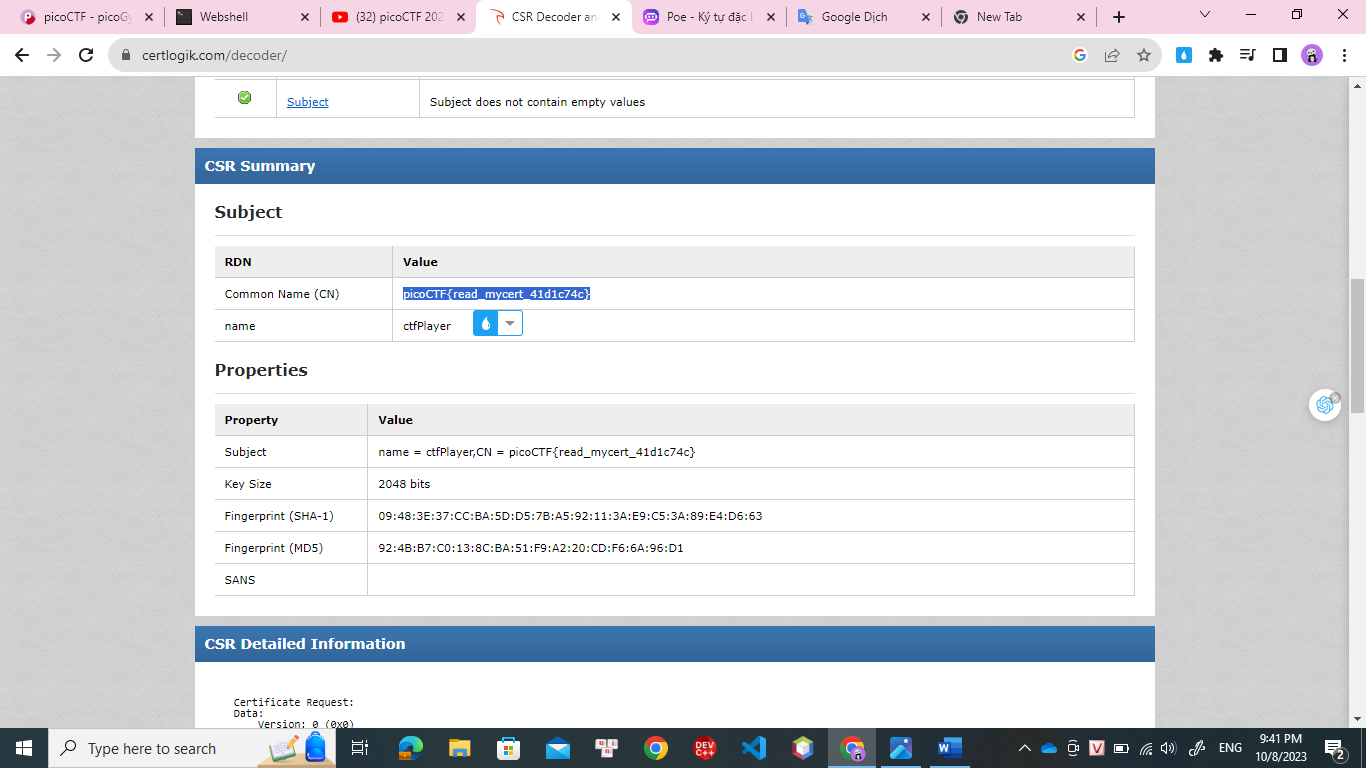
cat đọc dữ liệu đã được trích xuất trong file encrypted, decode dữ liệu đó bằng atbash cipher decode.



1. **ReadMyCert:**

File .csr là 1 yêu cầu ký chứng chỉ số được tạo ra từ 1 bên yêu cầu (web server hoặc ứng dụng) và gửi đến 1 tổ chức chứng thực (Certificate Authority – CA) để nhận được chứng chỉ số cho việc xác thực và bảo mật trên mạng. (Sử dụng public và private key)

Ở bài này lên google gõ csr file decoder rồi decode.



1. rotation

brute force caesar cipher