**Hướng dẫn sử dụng**

**Yêu cầu hệ thống**

1. Python 3.10 trở lên
2. Môi trường cài đặt pip

**Hướng dẫn cài đặt:**

Bước 1: Git clone ứng dụng:

Git clone <https://github.com/Huutkang/encryption-algorithm.git>

Bước 2 chuyển sang nhánh phuc:

Git checkout phuc

Bước 3: Cài đặt thư viên Pygame:

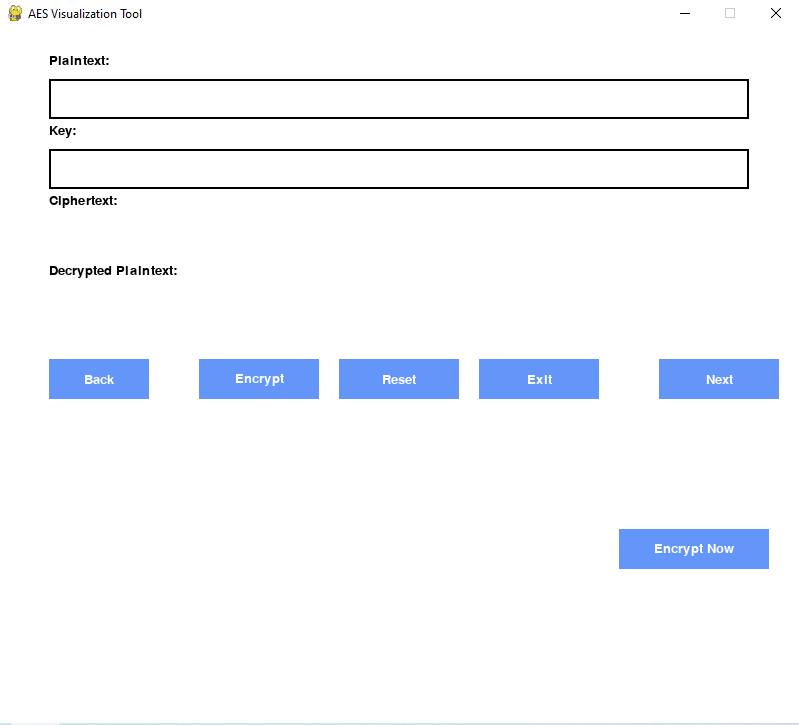
Pip install pygame

Bước 4: Chạy chương trình:

Python app.py

**Mô phỏng chương trình**

Đây là giao diện khi chạy chương trình:

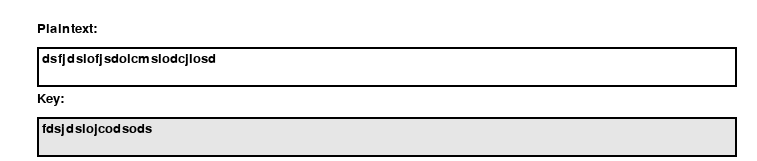


Trong đó: Plaintext và key là nơi nhập trường thông tin, plaintext là thông tin cần được mã hoá, key là thông tin đóng vai trò trong quá trình mã hoá và giải mã, ciphertext là kết quả sau khi mã hoá, còn decrypted plaintext là kết quả của việc mã giá trị ciphertext.

Các nút màu xanh trong giao diện có chức năng như sau:

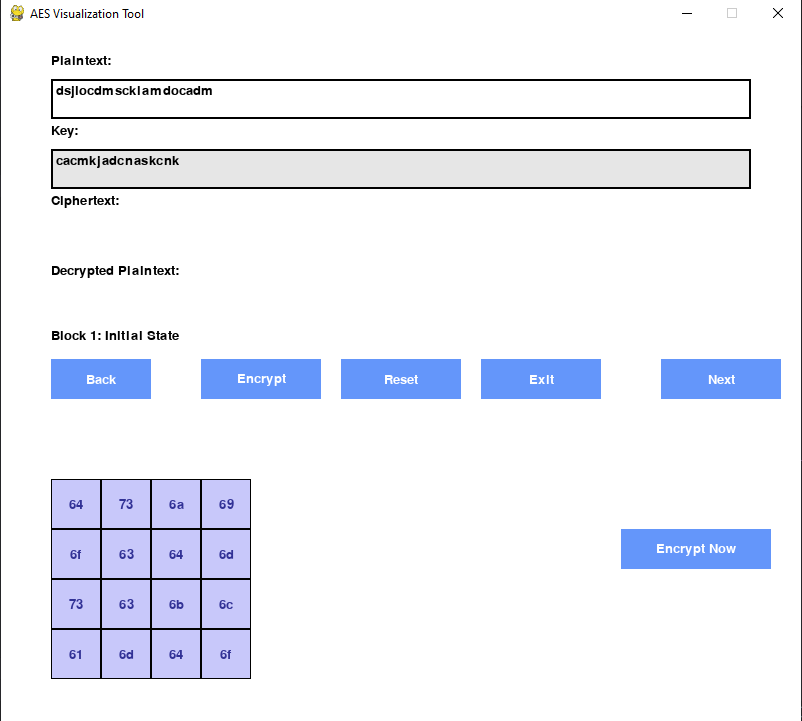
* Nút Encrypt thực hiện việc mã hoá, và nó sẽ tự động chuyển sang nút decrypt neues như thực hiện xong phần mã hoá.
* Nút Reset dùng để xoá toàn bộ các thông tin có trên giao diện và thực hiện một chu trình trực quan thuật toán mới.
* Nút exit dùng để thoát khỏi ứng dụng.
* Nút Back và Next dùng để theo dõi từng bước quá trình mã hoá và giải mã
* Nút Encrypt Now dùng để cho ra kết quả mã hoá trực tiếp thay vì theo dõi từng bước bằng nút Next, sau khi thực hiện xong bước mã hoá thì nút đó sẽ được chuyển thành nút Decrypt Now và nút đó có chức năng cho ra kết quả mã hoá ngay lập tức.

Tiến hành nhập giá trị bất kỳ cho plaintext và khoá



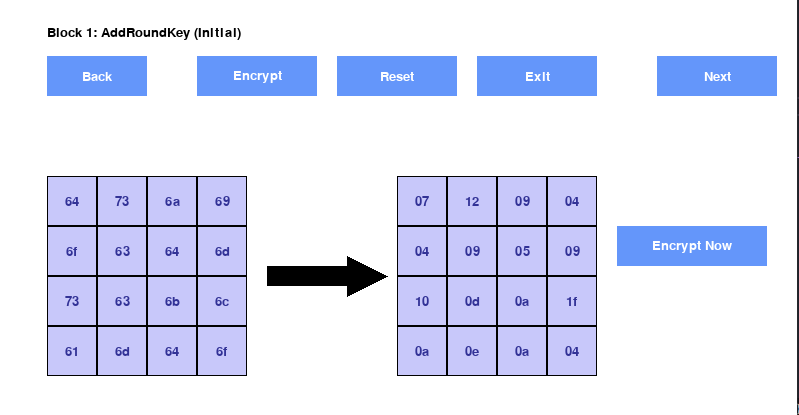
Ở đây giá trị plaintext không giới hạn số lượng ký tự nhập vào, còn giá trị key sẽ giới hạn 16 ký tự do tôi cho độ dài khoá là 128bit.

Sau khi nhập xong các trường thông tin trên, đến bước mã hoá. Nhấn nút Encrypt:

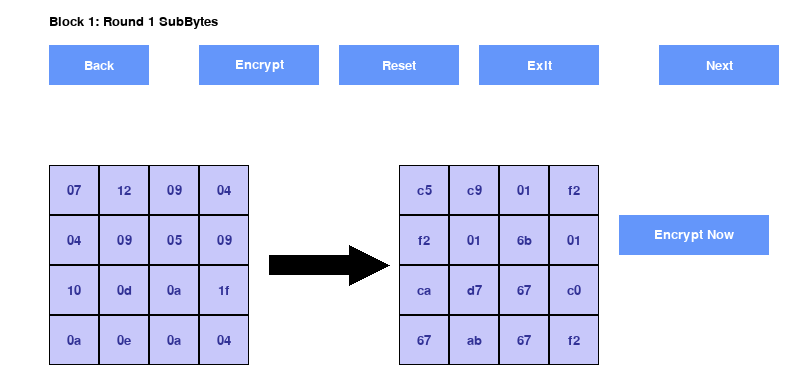


Sau khi nhấn nút Encrypt thì một khối ma trận 4x4 hiện ra thể hiện trạng thái đầu tiên của mã hoá, các thông tin được thể hiện như là Block 1 thể hiện thông tin của 16 ký tự đầu tiên của plaintext đã được chuyển dang dạng UTF-8 và thể hiện bằng ma trận. Ở đây nếu plaintext dài hơn 16 ký tự thì sẽ được chia ra thành các block để tiến hành mã hoá từng block, nếu block nào còn thiếu dữ liệu sẽ được padding thêm giá trị là số byte cần thêm để lấp đầy khối cuối cùng.

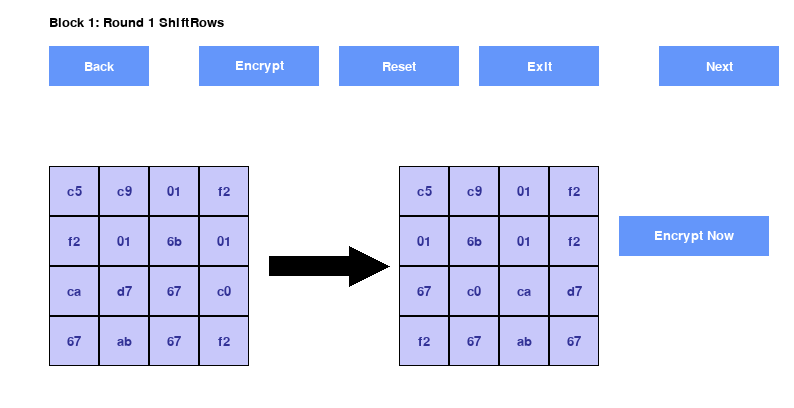
Tiếp tục nhấn next để theo dõi các bước của phần mã hoá:



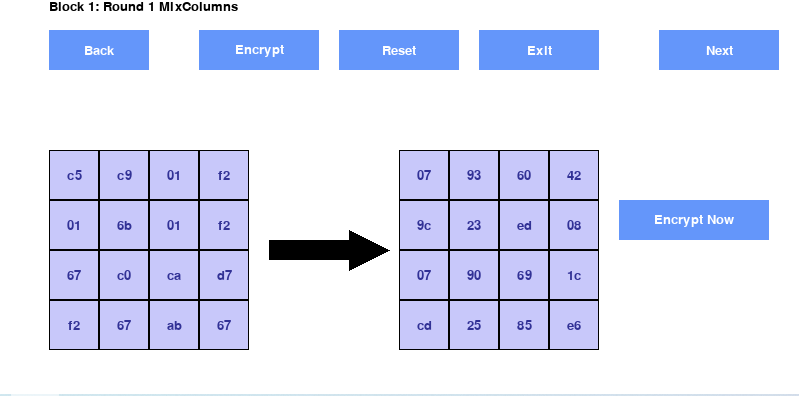
Bước add RoundKey



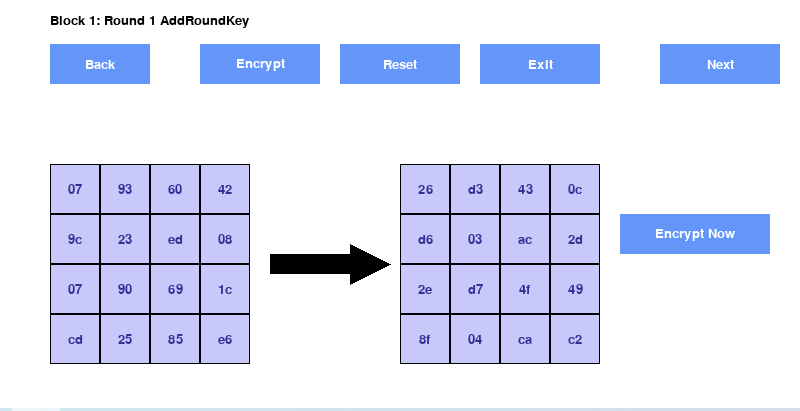
Bước SubBytes



Bước shift\_rows

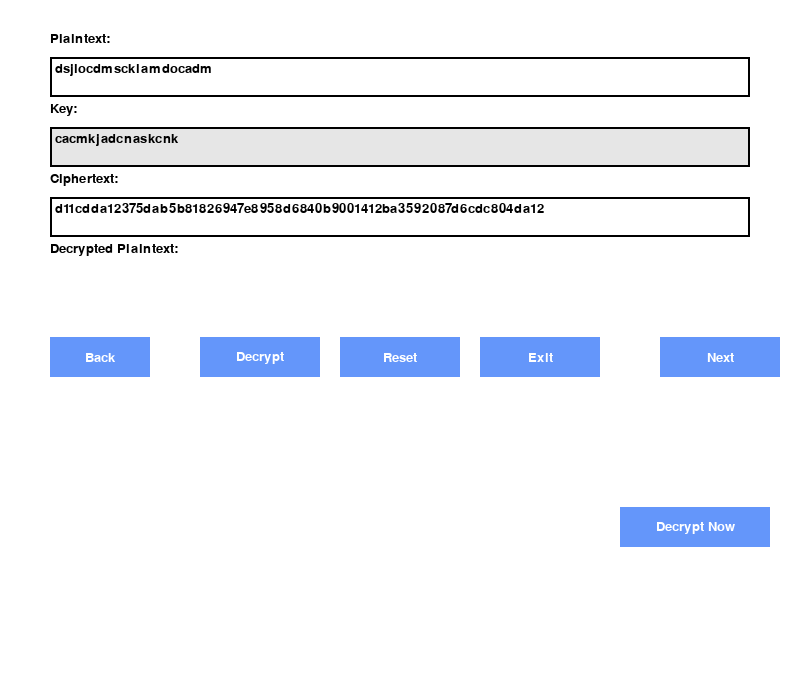


Bước mix\_column

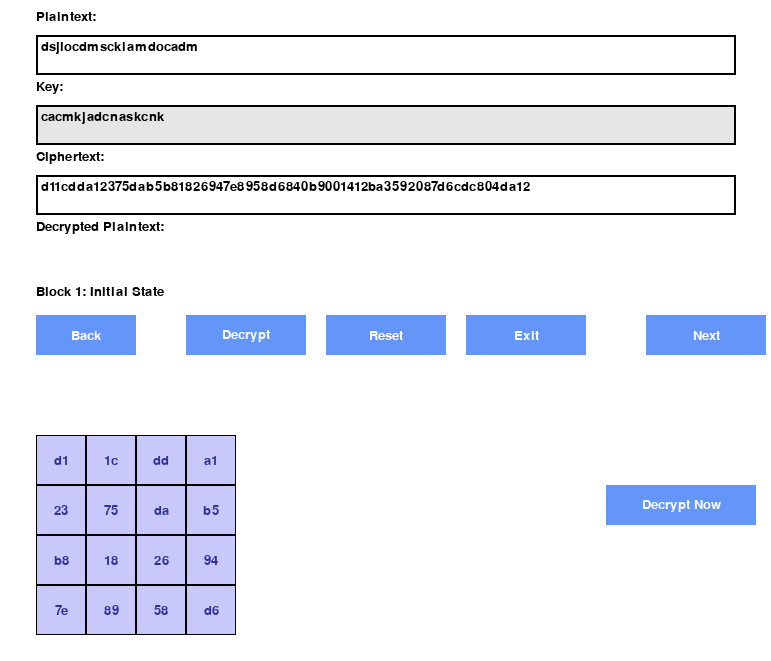


Bước Add\_round\_key

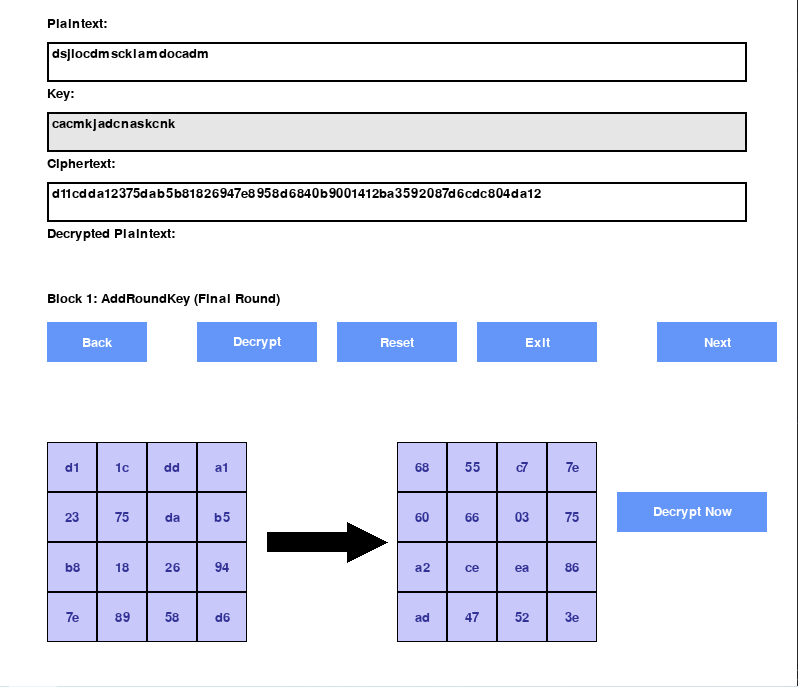
Theo quy trình của giải mã AES, sẽ có 9 bước thức hiện lặp lại một chu trình các phần như trên, đến bước thứ 10, thì sẽ bỏ phần mix\_column đi. Ngoài ra nút encrypt now sẽ giúp bạn cho ra kết quả của phần mã hoá luôn, đồng thời các nút encrypt và encrypt now được chuyển sang decrypt và decrypt\_now do đã thực hiện xong quá trình mã hoá.



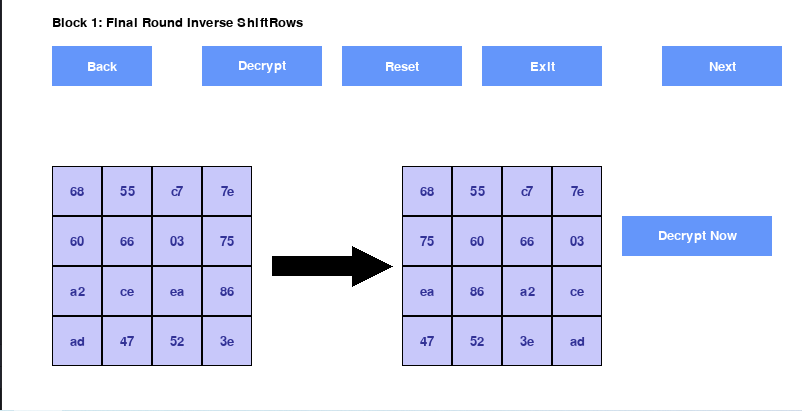
Tiếp tục đến phần giải mã, nhấn nút decrypt và theo dõi quá trình giải mã như sau:



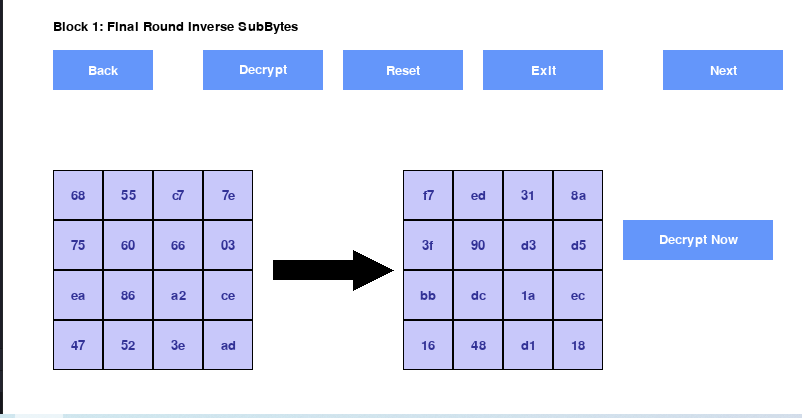
Lấy 16 khối byte trong ciphertext và đưa vào một một ma trận 4\*4



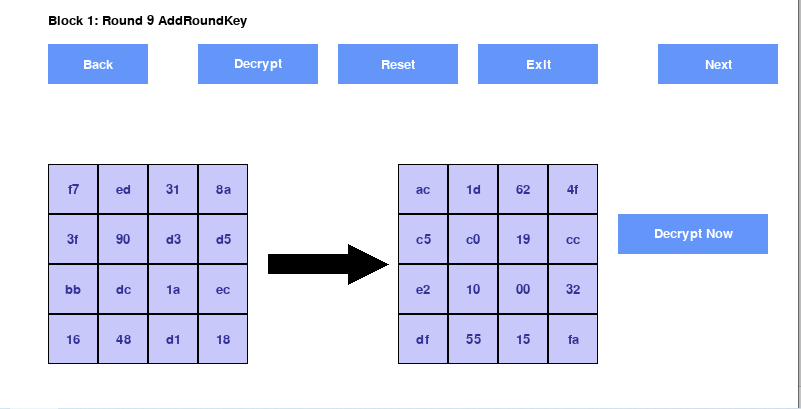
Tiếp đến là add\_round\_key,



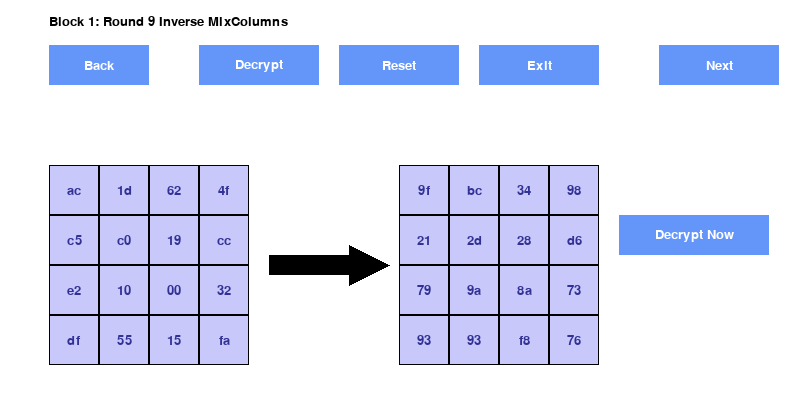
Bước inverse\_shift\_row



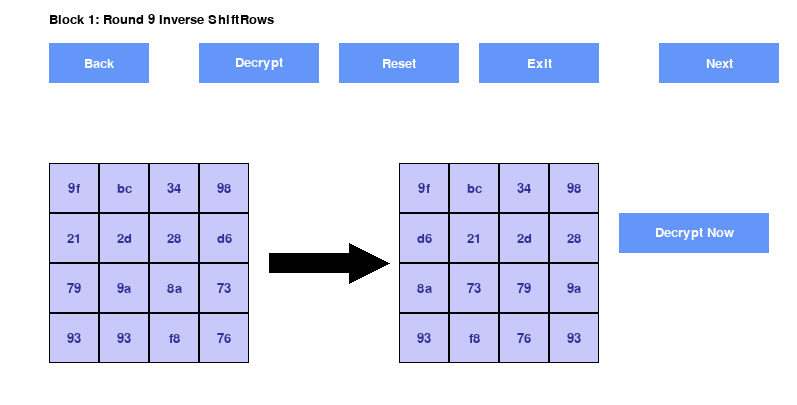
Bước inverse\_sub\_byte, do quá trình mã hoá là ngược lại của giải mã nên bước đầu tiên sẽ gồm có add\_round\_key, inverse\_shift\_row và inverse\_sub\_byte. Sau đó là thực hiện 9 lần liên tiếp một chu trình gồm các phần như sau:



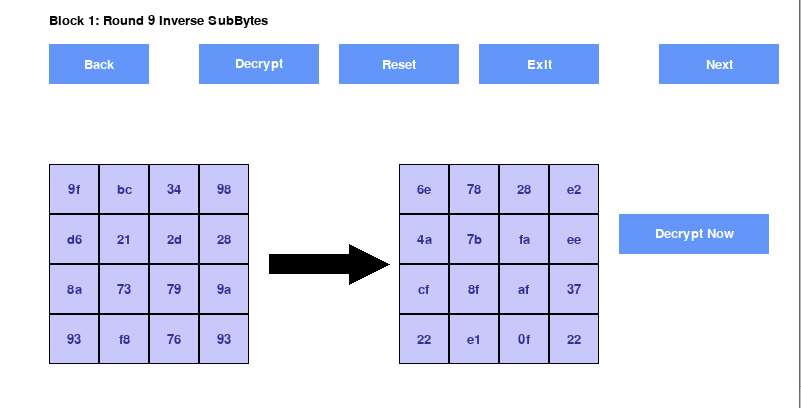
Add\_round\_key



Inverse\_mix\_column



Inverse\_shift\_row



Inverse\_sub\_byte

Chạy hết 9 vòng, ta đến bước cuối cũng là add\_round\_key, sau đó kết quả thu được là một plaintext giống với giá trị ban đầu nhập vào. Bạn có thể sử dụng nút decryptnow để cho ra kết quả luôn quá trình giải mã

