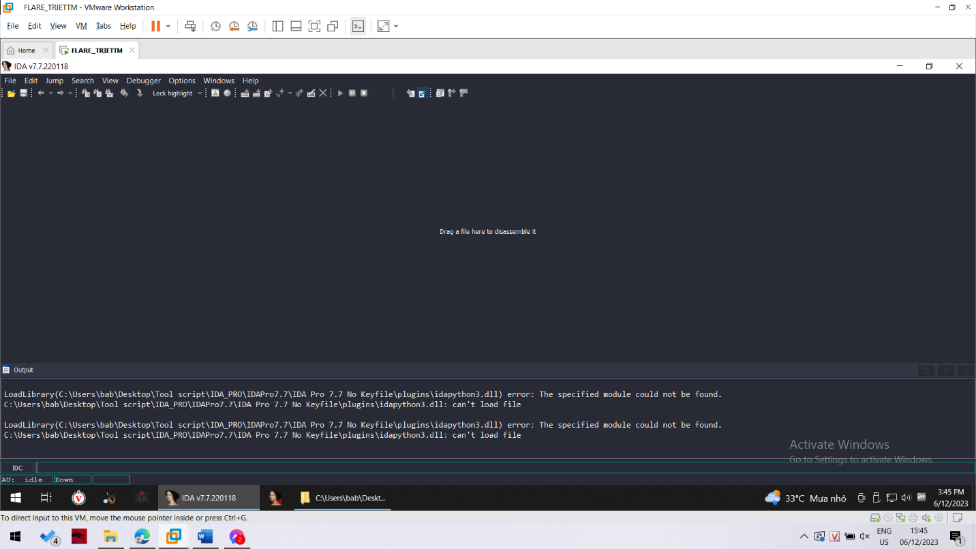
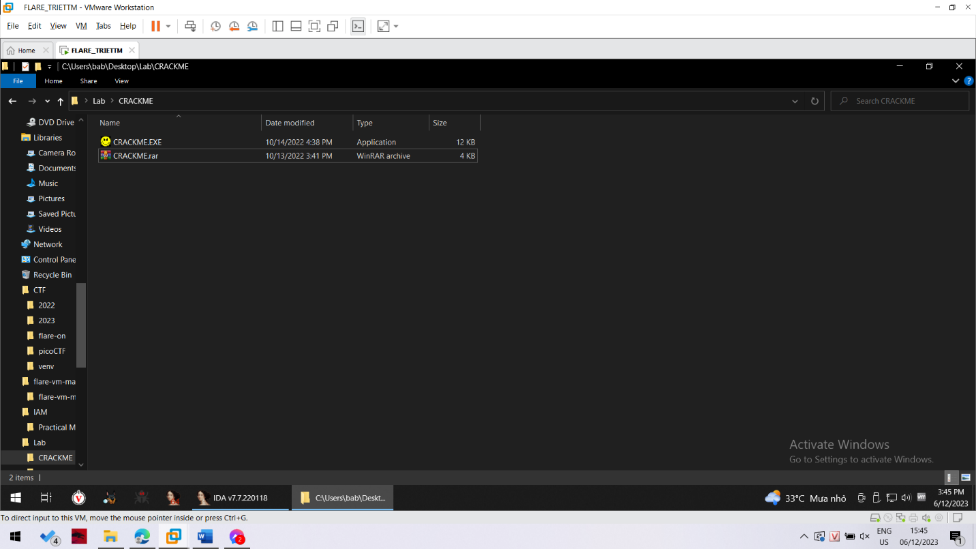
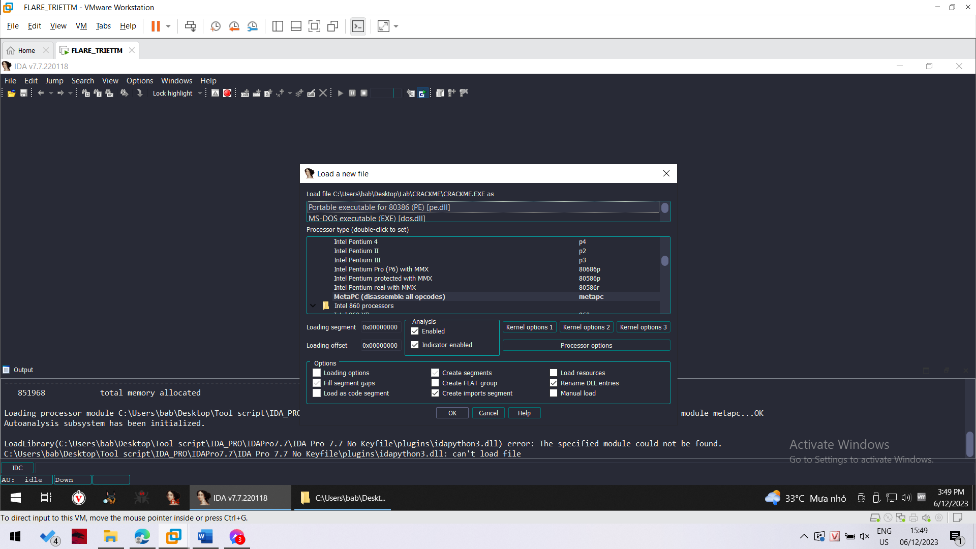
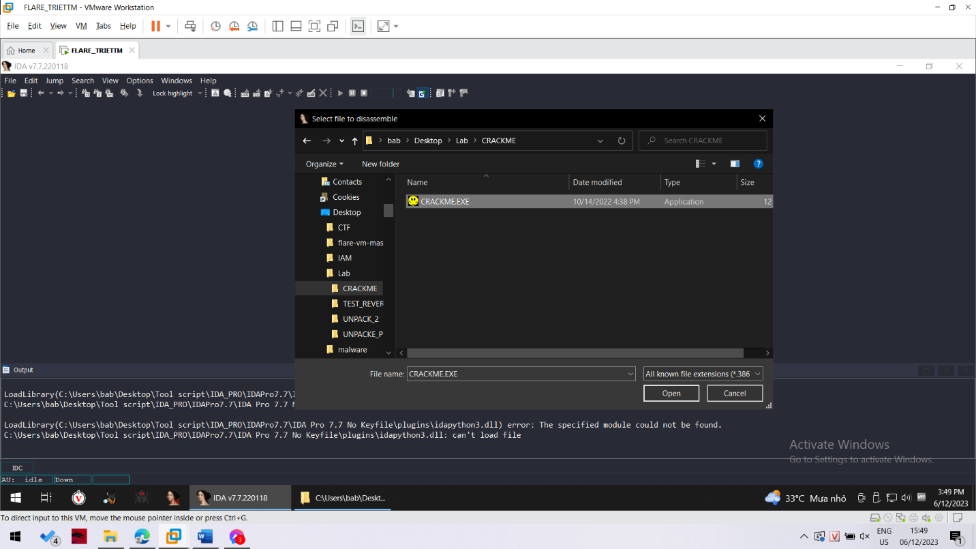
**Lab 9.2: Disassembling C on Windows**

Ở bài lab này chúng ta sẽ tiến hành phân tích một chương trình được viết bằng C trên hệ điều hành Windows.

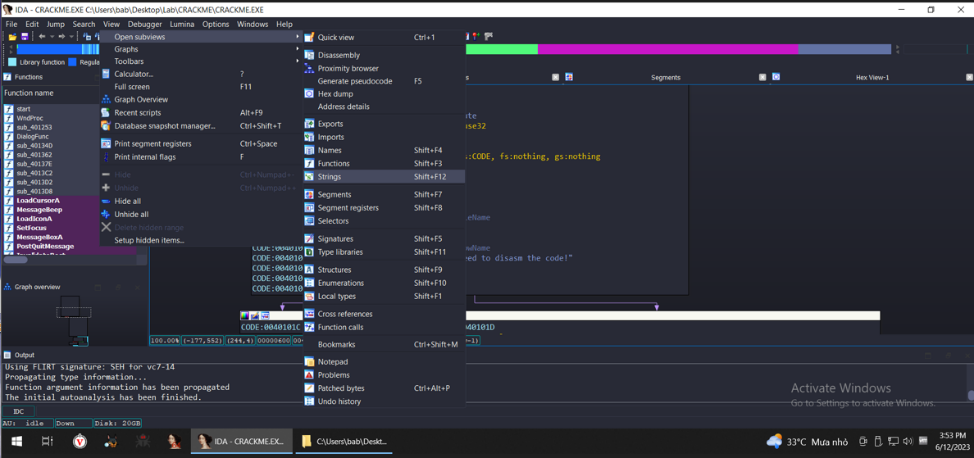
Đầu tiên ta mở công cụ IDA pro đã tải ở phần trước lên rồi load file PE CRACKME mà em đã chuẩn bị trước vào.



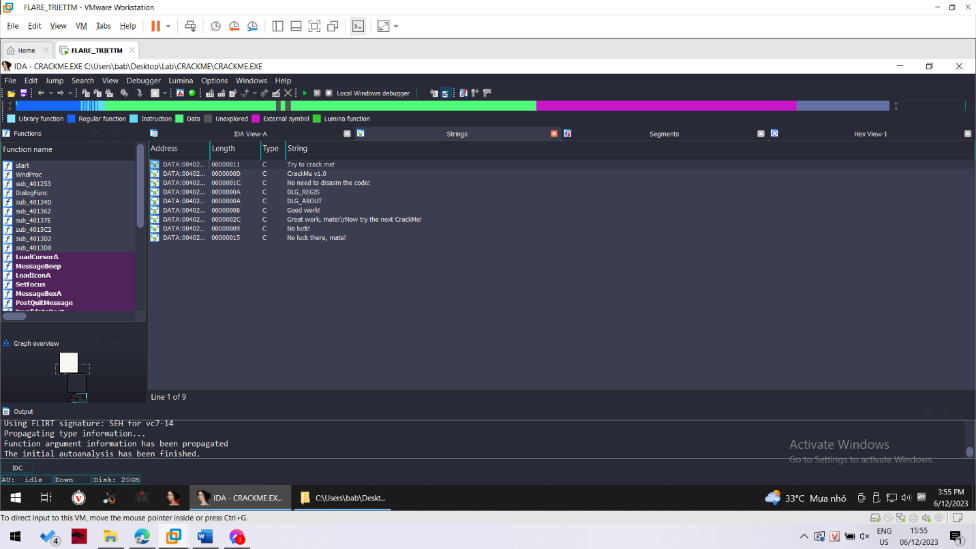




Chọn OK để IDA tiến hành phân tích, extract ra string, disassembler ra asm code, phân tích các dll được gọi cùng với rất nhiều thứ khác nữa.



Ta vào phần View -> Open Subview rồi chọn String. Tại đây ta có thể xem tất cả các đoạn chuỗi có trong chương trình này do IDA phân tích ra.



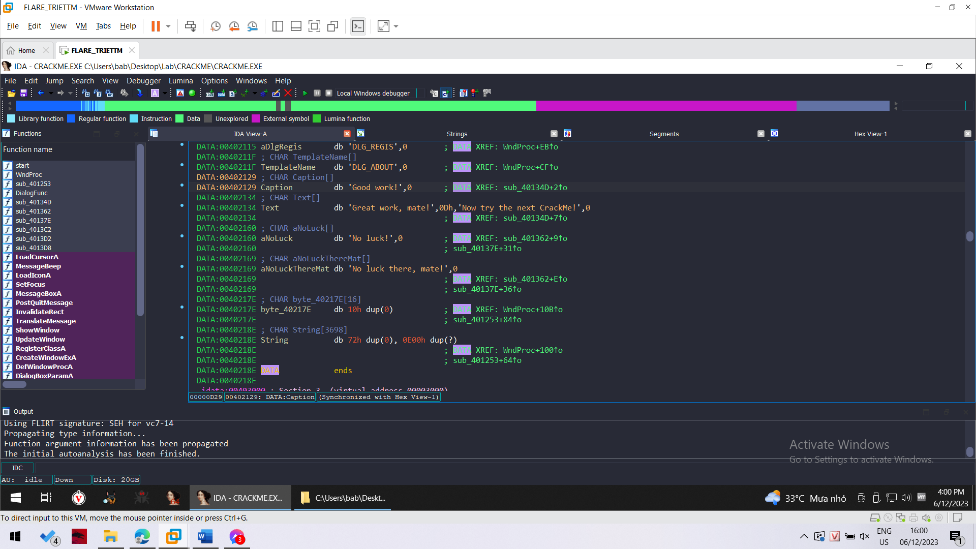
Khi phân tích bất cứ chương trình nào việc đầu tiên ta nên làm là xem strings của nó, việc tìm kiếm các đoạn string khả nghi sẽ giúp ta nhanh chóng tìm được vị trí code quan trọng nhanh hơn, tiết kiệm được nhiều thời gian. Ở đây ta có thể thấy có 4 đoạn chuỗi khả nghi dưới cùng lần lượt là:

Good work!

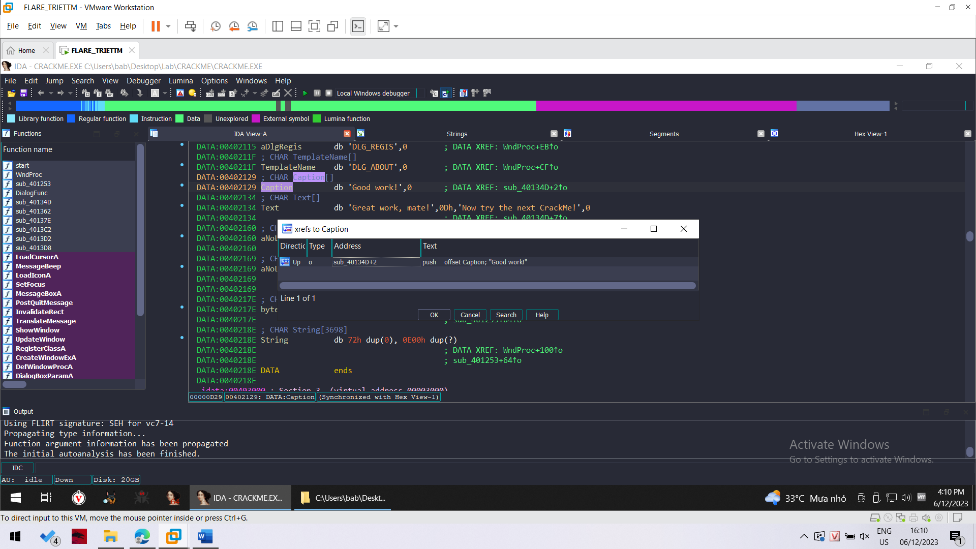
Great work, …

No luck

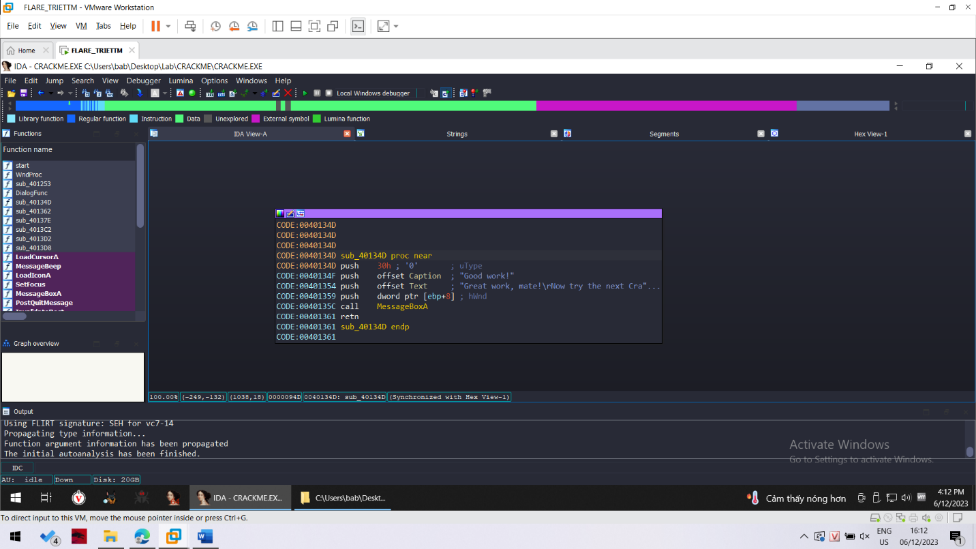
Double click vào các chuỗi này sẽ dẫn ta đi đến nơi chứa chuỗi trong vùng data.



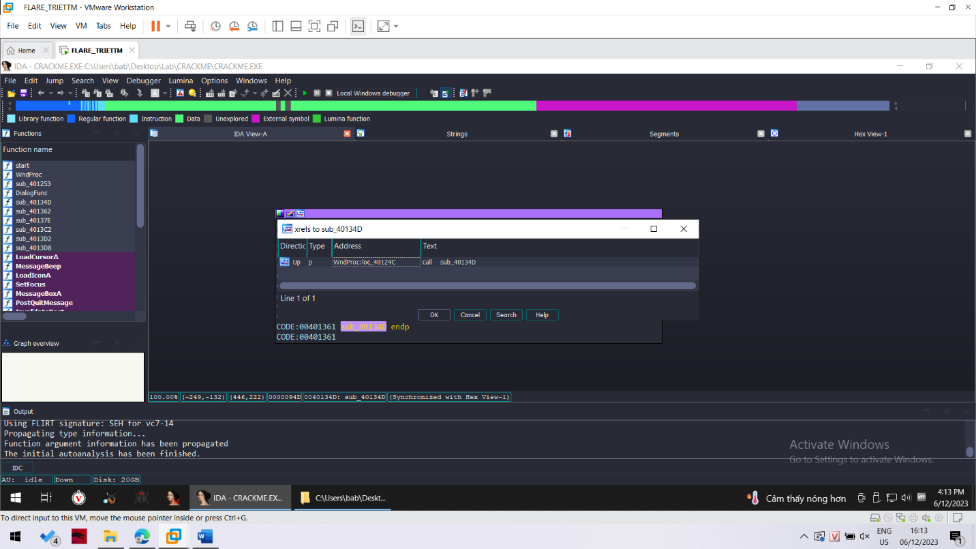
4 đoạn chuỗi này được chứa trong section DATA, ta có thể tìm nơi tham chiếu đến các đoạn chuỗi này bằng cách sử dụng công cụ reference của IDA với phím tắt là x để xem các tham chiếu.

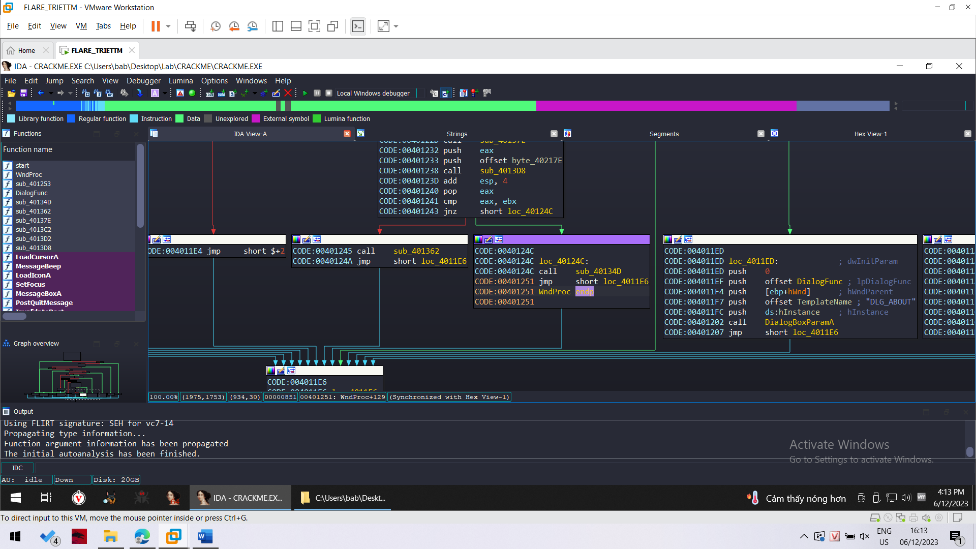


Ta thấy chuỗi “Good work!” được tham chiếu bởi hàm sub\_40134D, từ đây ta có thể double click vào nó, IDA sẽ nhảy tới ngay các instruction gần chuỗi này. Khi đó ta sẽ phân tích kỹ hơn xung quanh đoạn string này thôi.



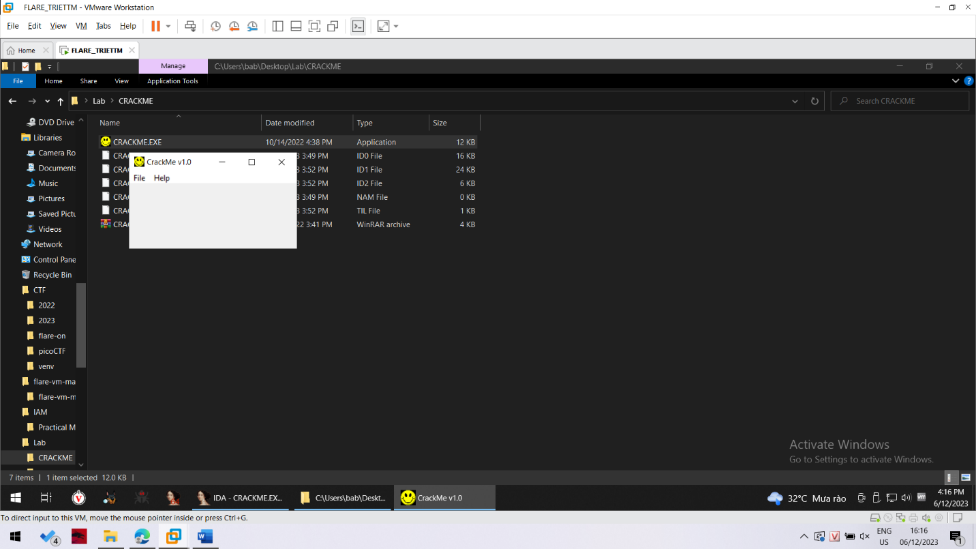
Bấm x vào hàm này để xem hàm này được gọi ở đâu.



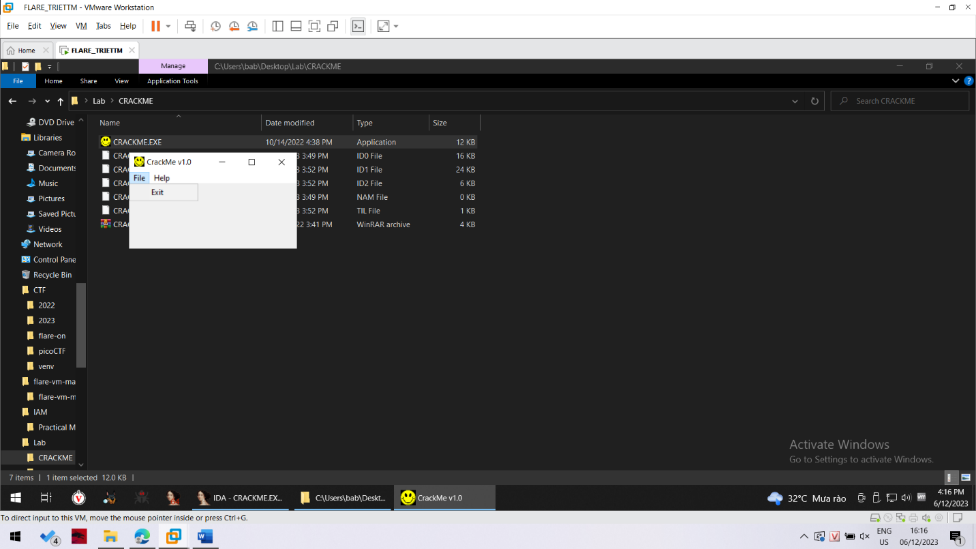


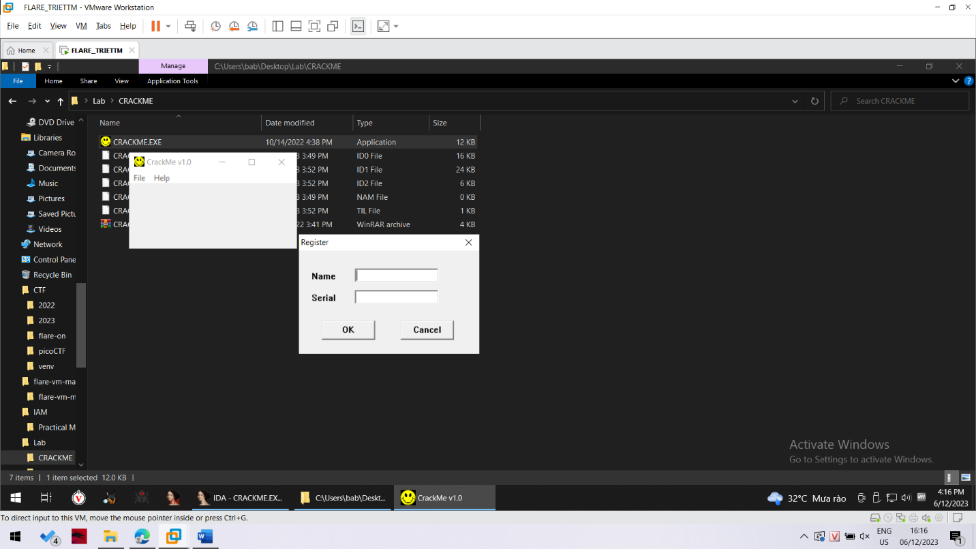
OK vậy là tới được một hàm khá là lớn rồi, có lẽ chỗ này là nơi chương trình thực sự chạy.

Bây giờ có hơi trễ một chút mà có lẽ ta nên chạy thử chương trình này một chút.



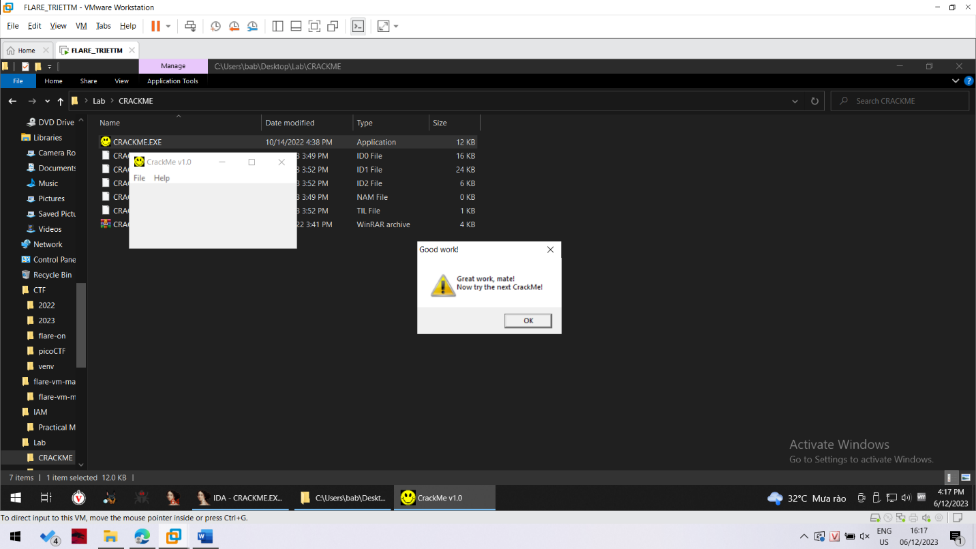
Một màn hình CRACKME hiện ra.

Nhấn file thì ra Exit

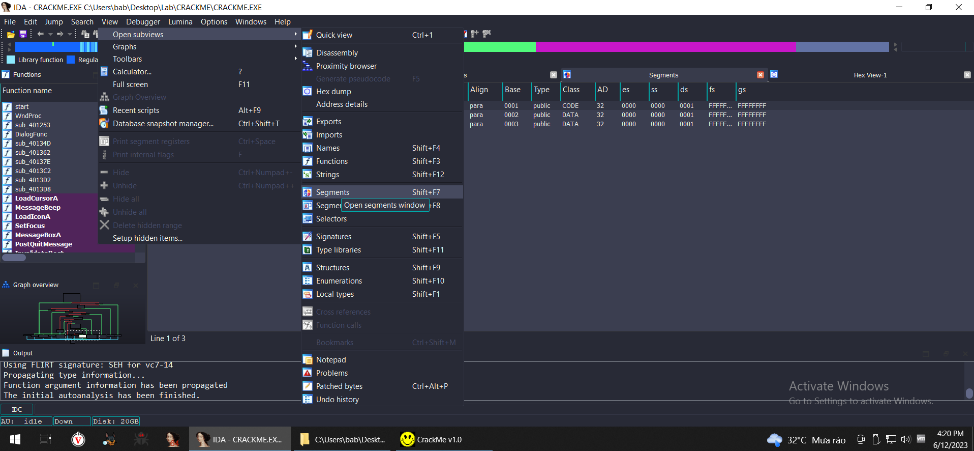


Bấm chữ Register thì sẽ ra một cửa sổ đăng nhập yêu cầu name và pass.

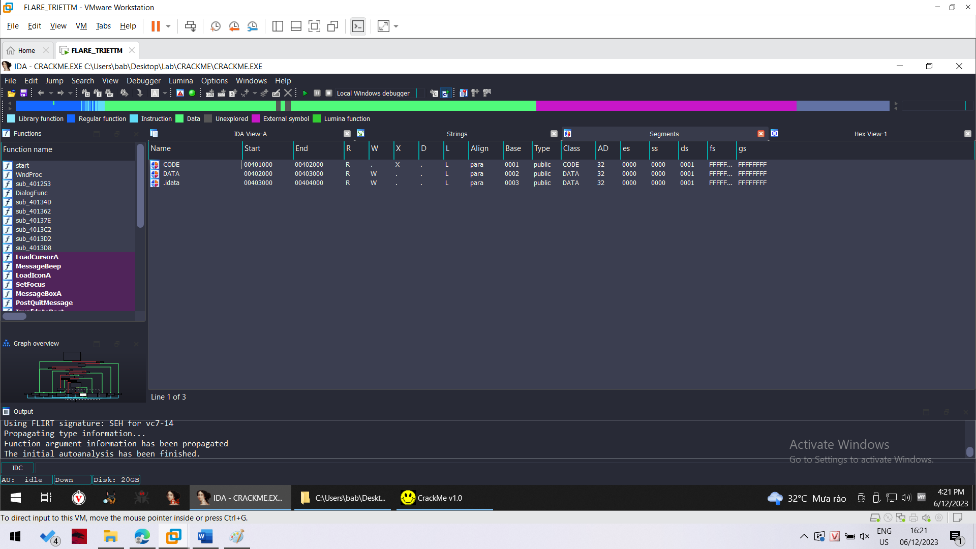
Ta nhập thử đại rồi bấm OK thì sẽ nhảy ra dòng chữ như sau



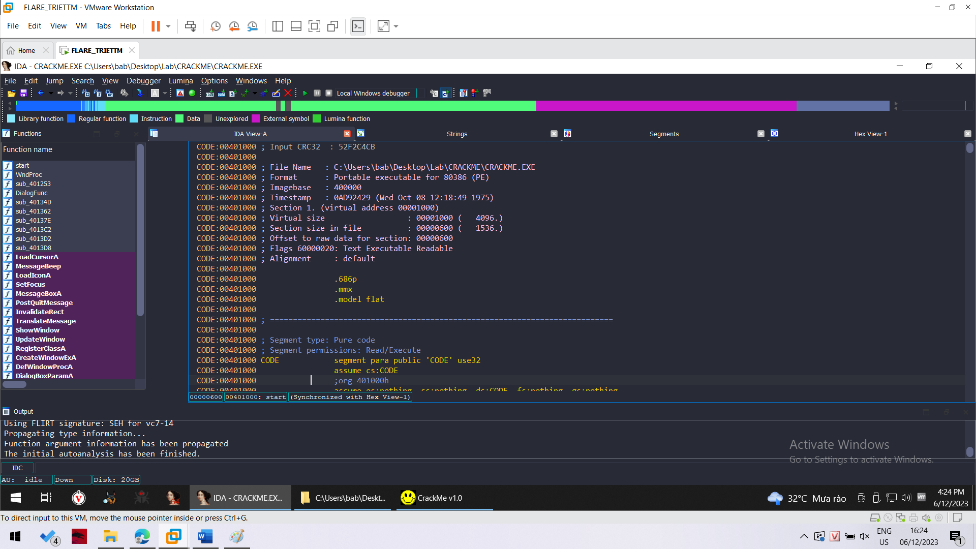
Đây ta có thể thấy dòng chữ khả nghi như khi xem strings. Rất có thể những dòng chữ này dùng để thông báo xem coi ta có nhập đúng name và pass hay chưa.



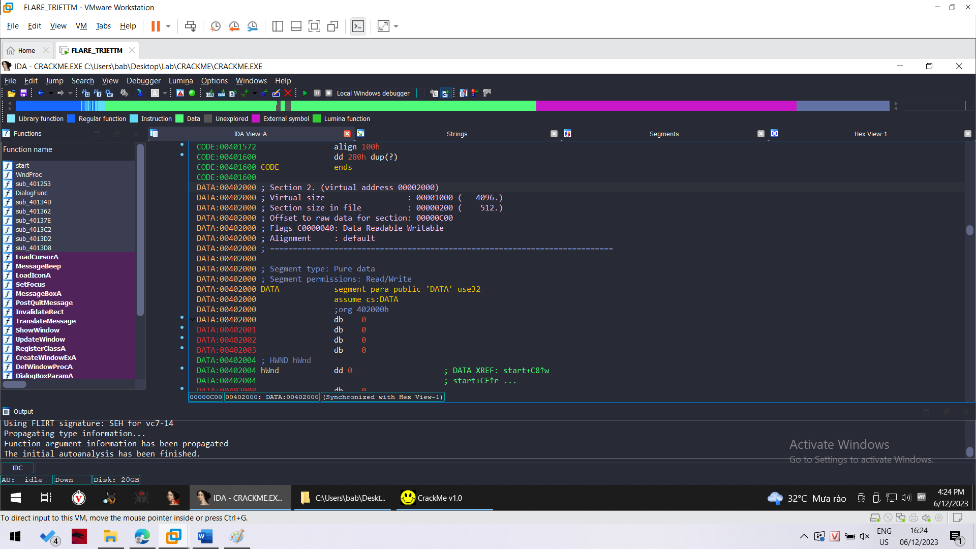
Quay lại với IDA, bây giờ ta sẽ vào View -> Open Subview -> Segments, để xem các trường có trong chương trình.



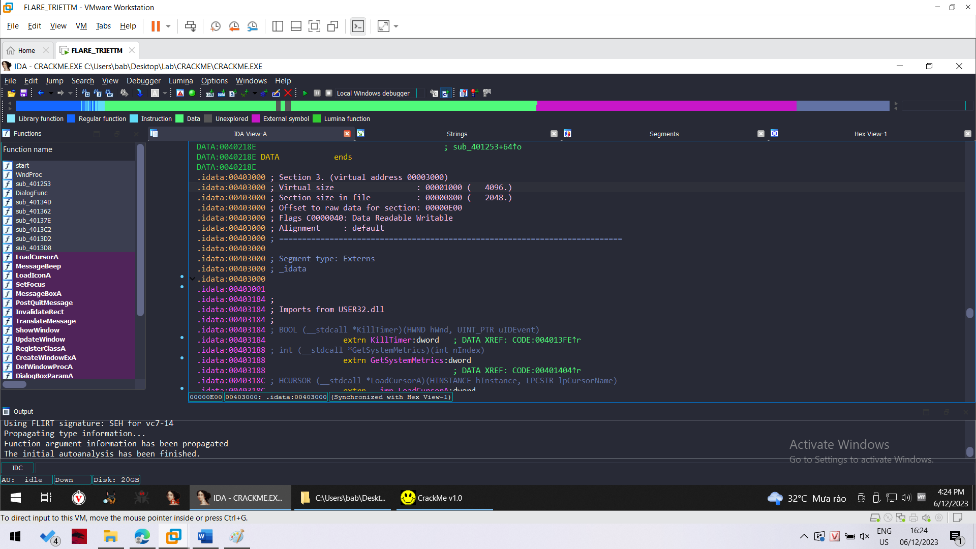
Ở đây có ba vùng, một là vùng CODE chứa code asm của chúng ta để chương trình chạy, hai là vùng DATA nơi chứa các đoạn chuỗi khả nghi như ta thấy lúc nãy và ba là vùng idata nơi chứa bảng IAT(import address table) nhằm chỉ định vùng nhớ tham chiếu tới các thư viện DLL, các data khác cần dùng để chạy file exe này.



Vùng CODE



Vùng DATA

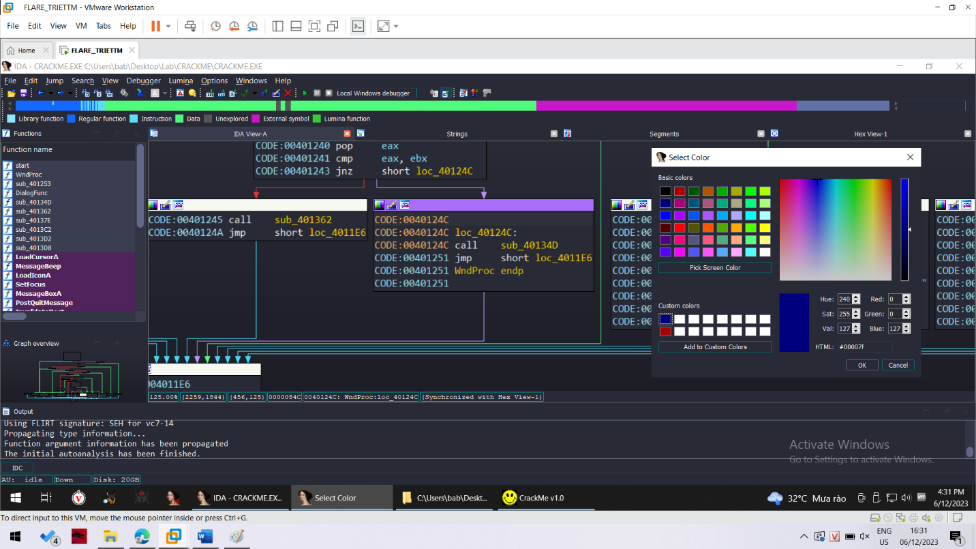


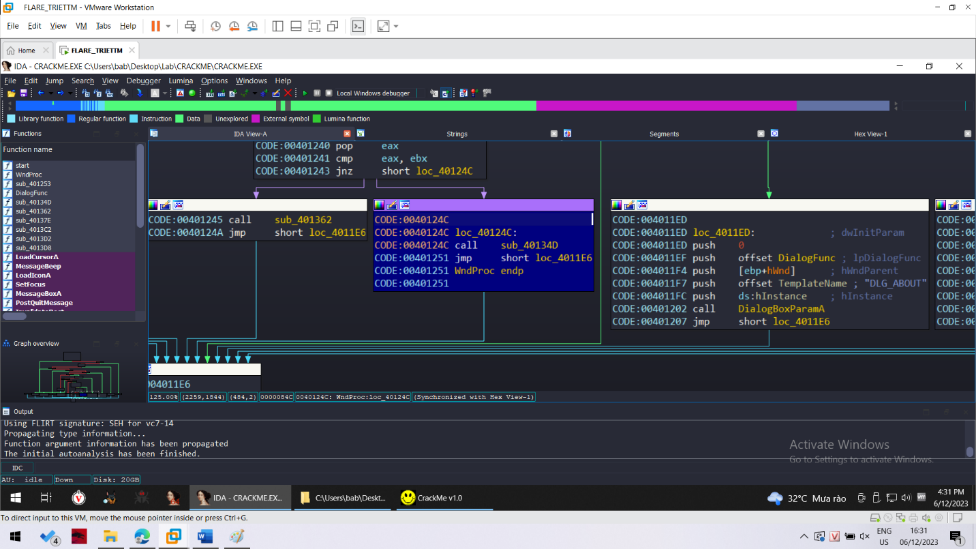
Vùng idata

Đặc biệt ở vùng code ta có thể linh hoạt chuyển đổi giữa graph view và text view thông qua shortcurt tab tùy vào sở thích đọc code của mỗi người.

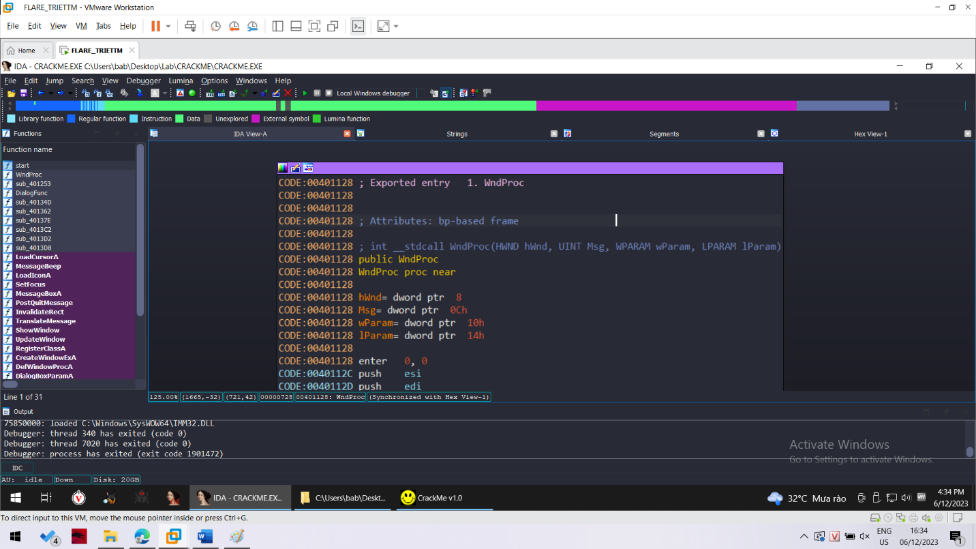
Quay lại với vị trí khi nãy ta tìm ra chuỗi Great Work!

Dựa vào kết quả chạy được khi chạy thử chương trình ta có thể đánh dấu đoạn code này màu xanh, nghĩa là trạng thái ta cần đạt tới.

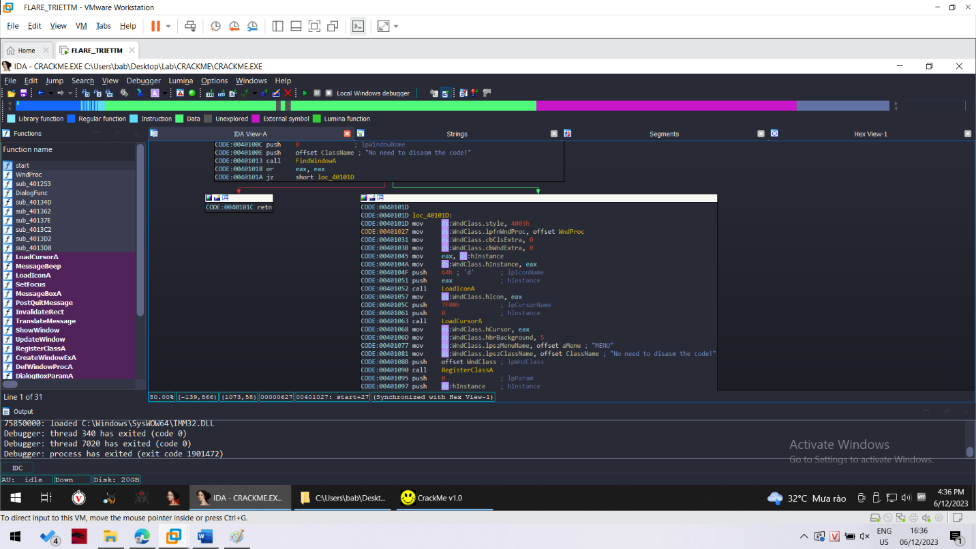
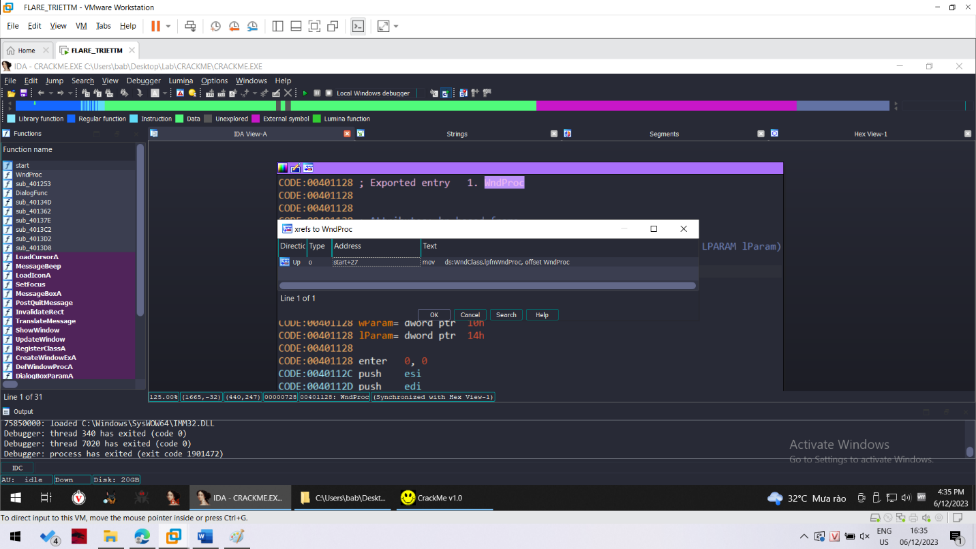




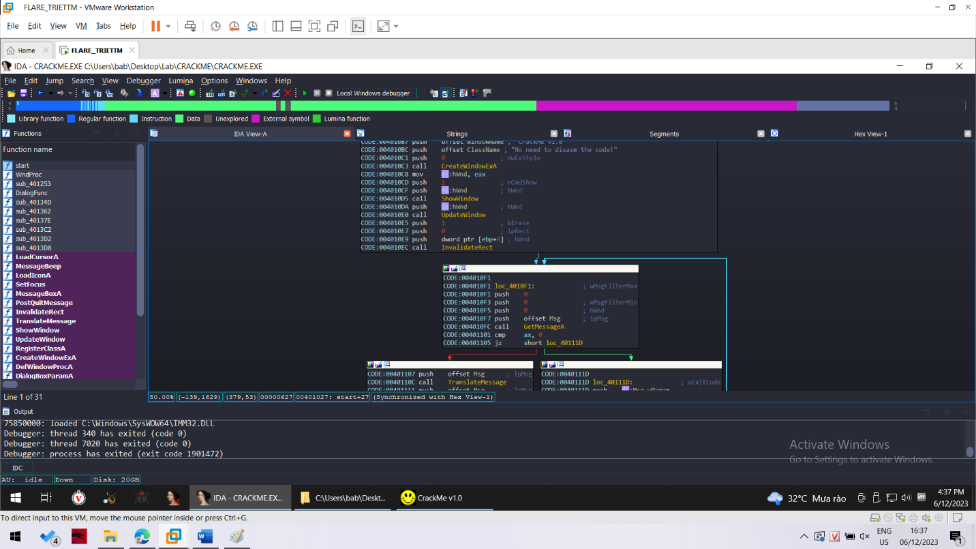
Kéo lên tới đầu hàm này ta thấy hàm này tên là WndProc



Ta thử xem hàm này được tham chiếu bởi đâu.

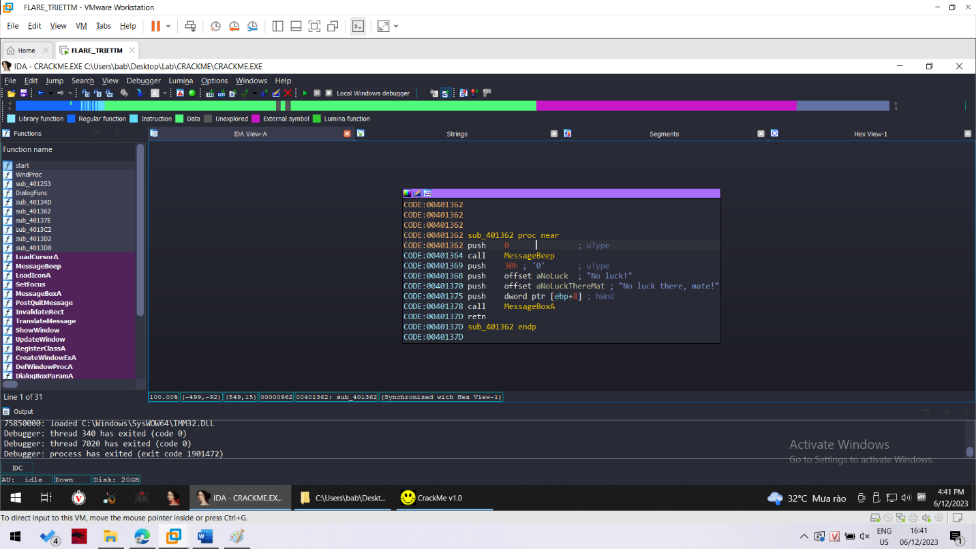


Quao đến đây ta có thể tìm thấy một số hàm như FindWindow dùng để tạo ra khung cửa sổ khi ta chạy chương trình, hàm LoadIcon để load icon hình mặt cười của chương trình.



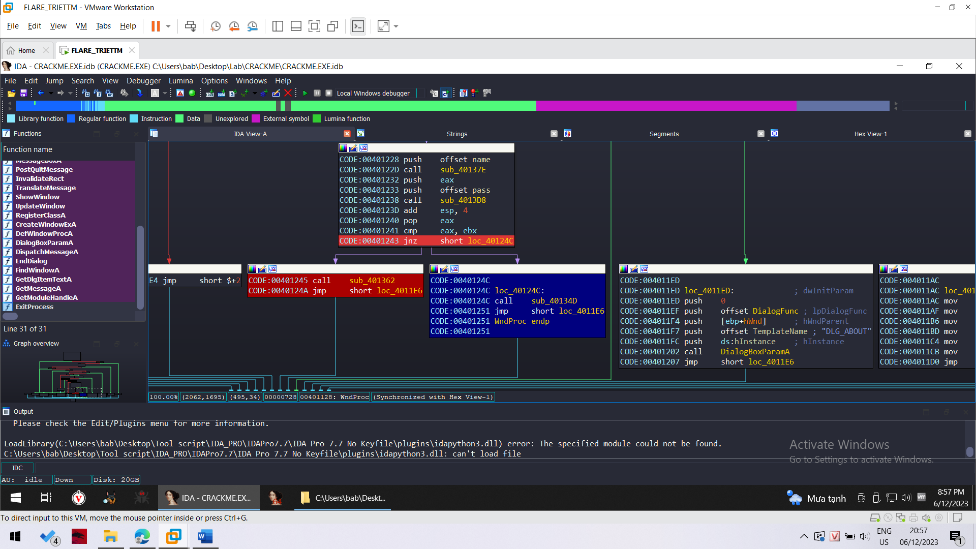
Một số hàm tuong tác với khung cửa sổ nhằm nhập các input của ta như name và serial để trigger các điều kiện đúng sai.

Bây giờ ta quay lại với đoạn code trước, tại đây bên cạnh đoạn code ta đánh dấu xanh thử xem đoạn code còn lại như thế nào

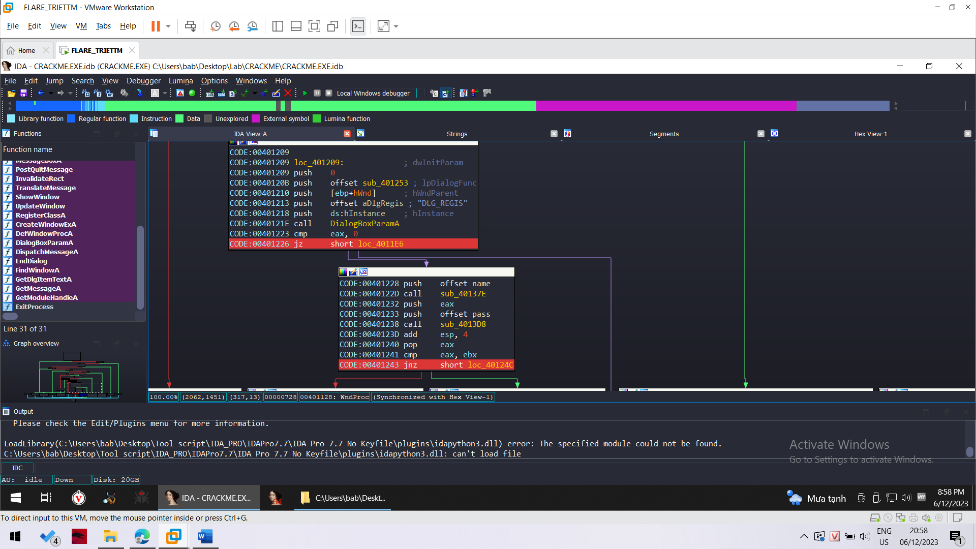


Ta thấy ở đây các chuỗi No luck được push lên stack rồi sau đó gọi các hàm MessageBox để tạo ra khung cửa sổ thông báo nhập name và pass sai.

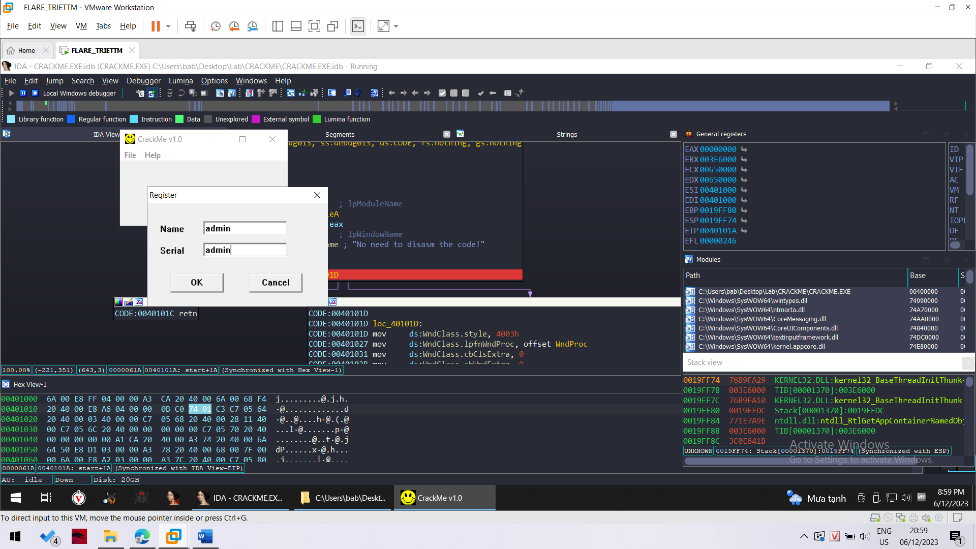
Chính vì vậy ta sẽ đánh dấu đoạn code này bằng màu đỏ.



Vậy thì ở đây sẽ có một đoạn code dùng để check xem name và pass coi đúng hay chưa rồi tùy vào kết quả mà rẽ sang hai hướng đi đúng hoặc sai.

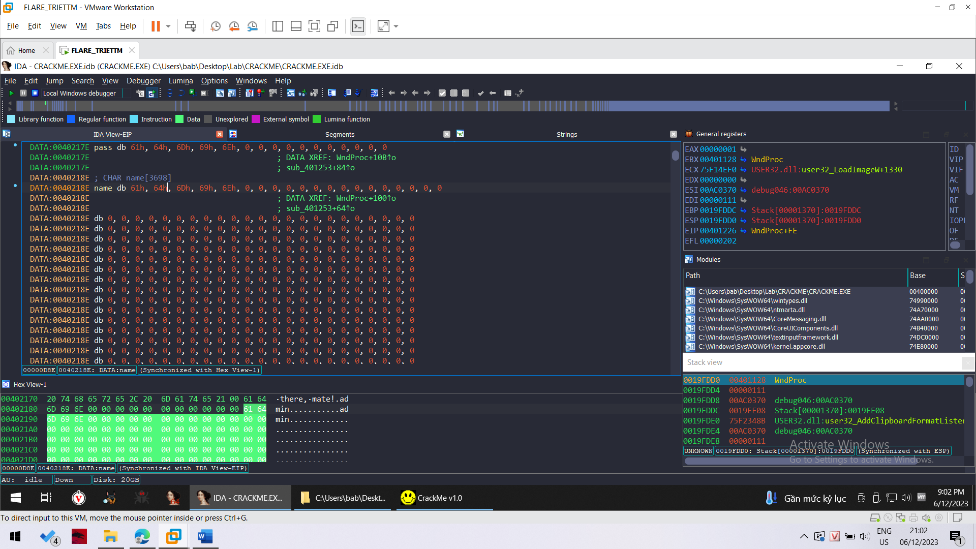


Em tiến hành đặt thử 2 breakpoint như hình trên để theo dõi input từ bàn phím

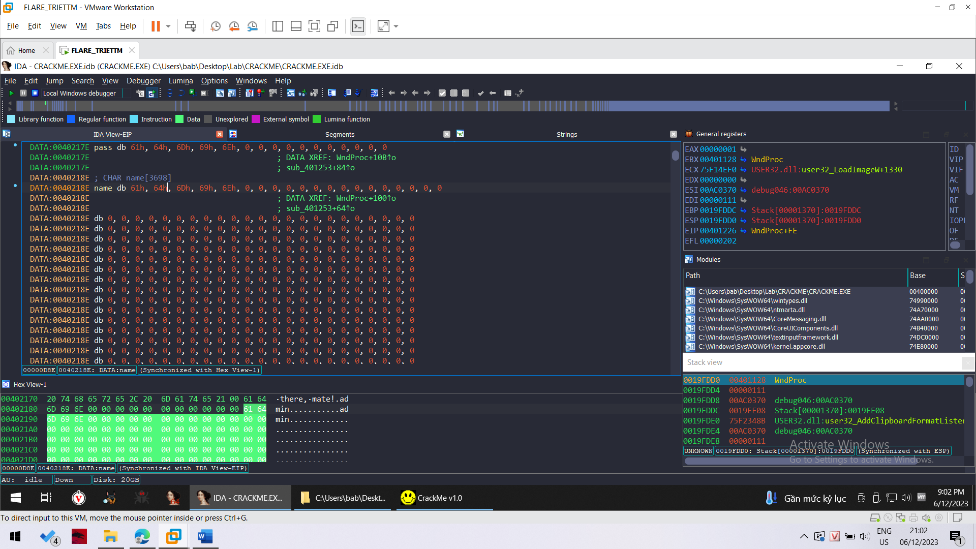


Nhập thử username và pass lần lượt là admin và admin

Trong quá trình runtime ta có thể thấy chuỗi name admin được lưu lào vùng nhớ này

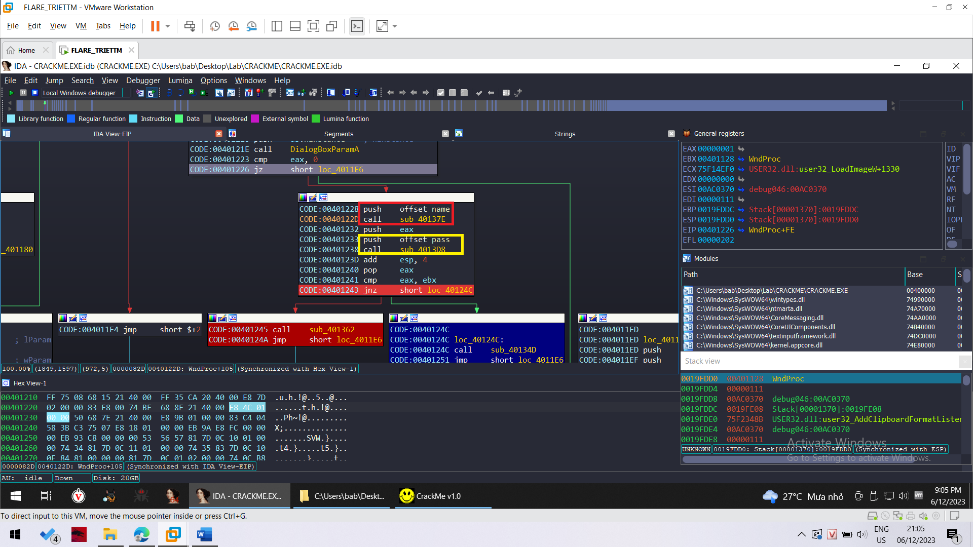


Còn chuỗi password thì được lưu ở vùng nhớ này

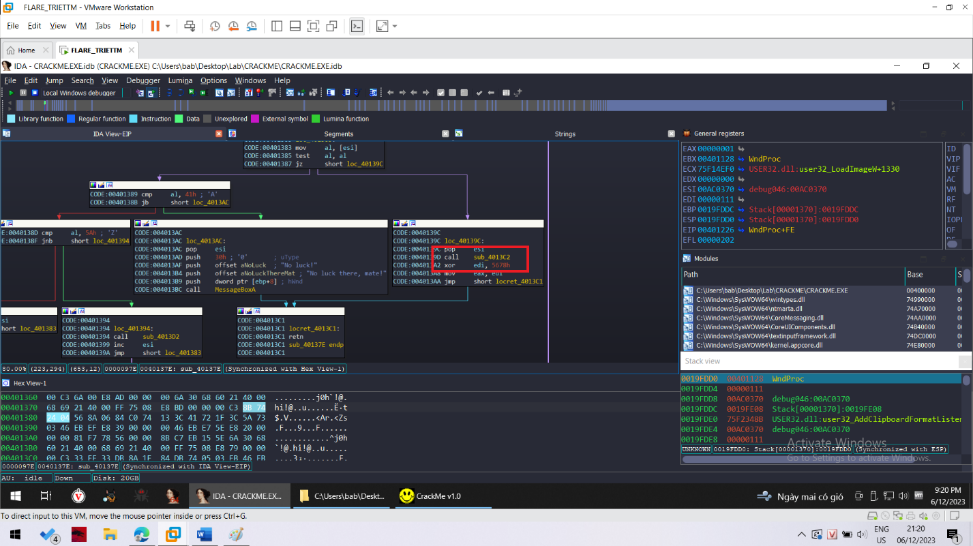


Chương trình sẽ xử lý 2 tham số này một cách riêng biệt.

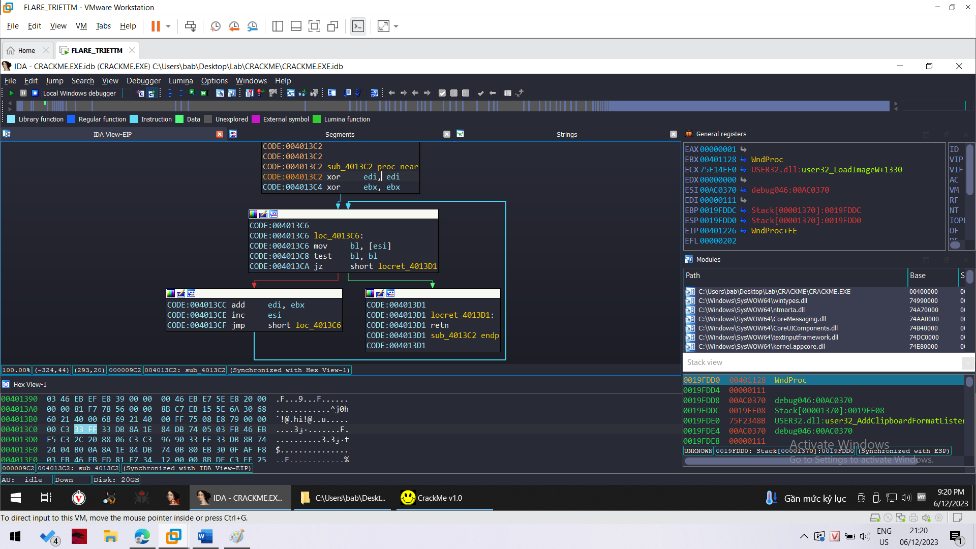
Name thì được xử lý trong hàm sub\_40137E trả ra kết quả ở thanh ghi eax, sau đó eax được push lên stack

Pass thì được xử lý trong hàm sub\_4013D8 trả ra kết quả ở thanh ghi ebx

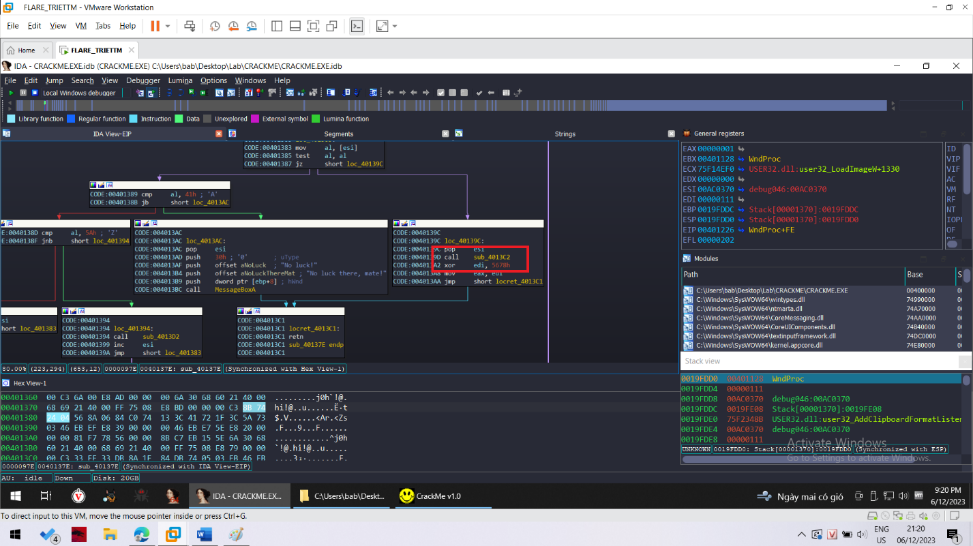
Đối với đoạn code xử lý name, đầu tiên nó sẽ đi qua từng ký tự của name và check xem ký tự đó là lowercase hay không, nếu có thì uppercase nó lên.

Sau đó đi vào một hàm tính toán chính.

Nhấp vào hàm sub\_4013C2 này.

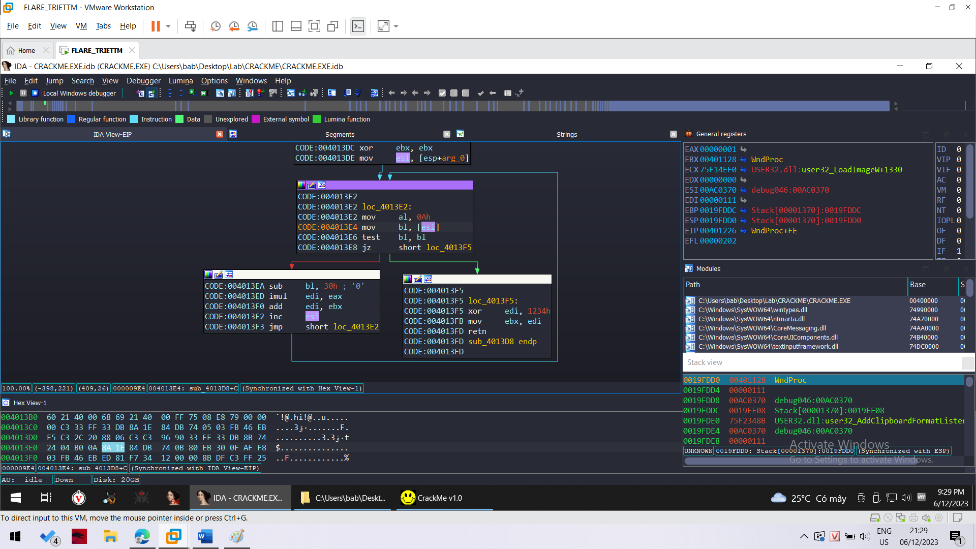


Ở hàm này nó sẽ đem từng ký tự của name (hiện tại là admin) lưu vào thanh ghi bl sau đó cộng giá trị decimal của tất cả ký tự này lại với nhau thành một tổng rồi trả về trong thanh ghi edi.

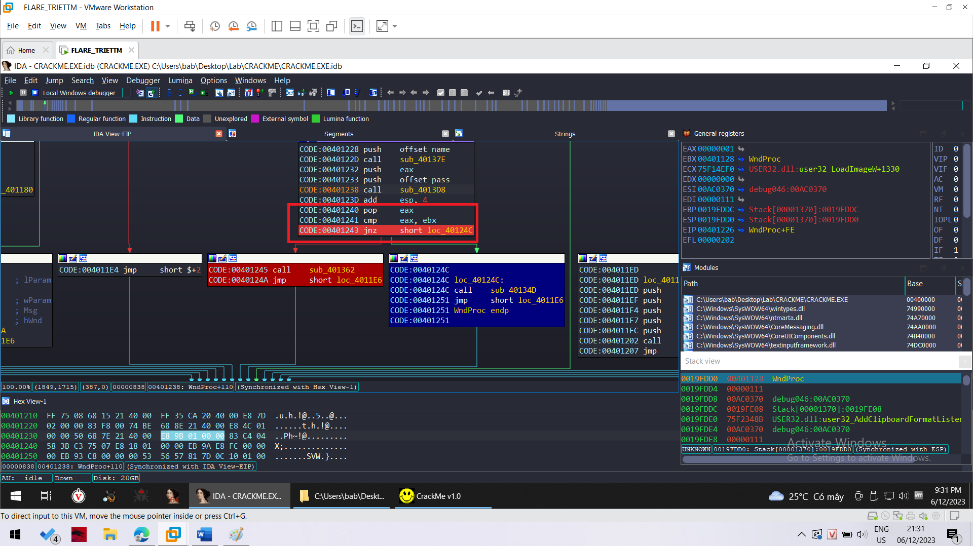


Về lại hàm trước, thanh ghi edi sau khi chứa tổng các ký tự của chuỗi năm này thì được đem đi xor với 0x5678 rồi trả về kết quả trong thanh ghi eax.

Quay sang đoạn code xử lý password

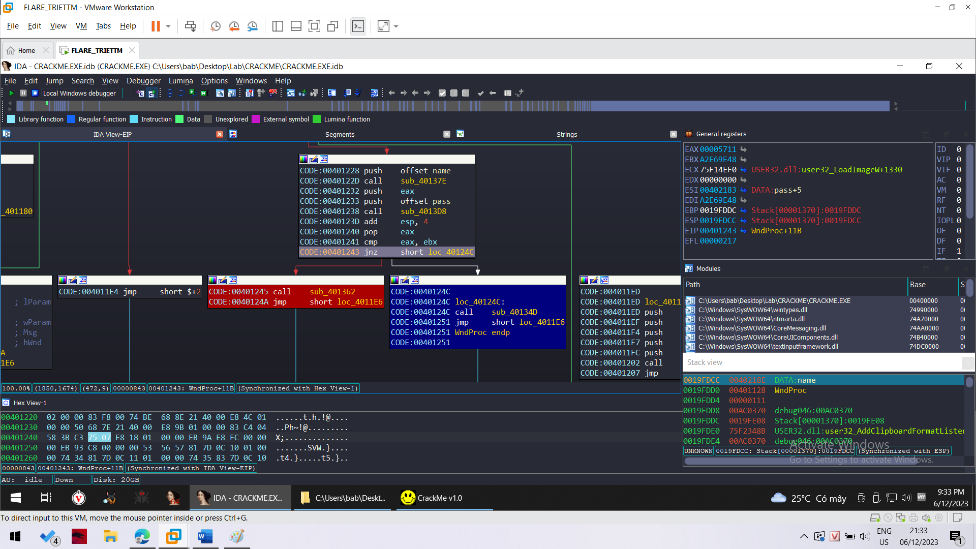


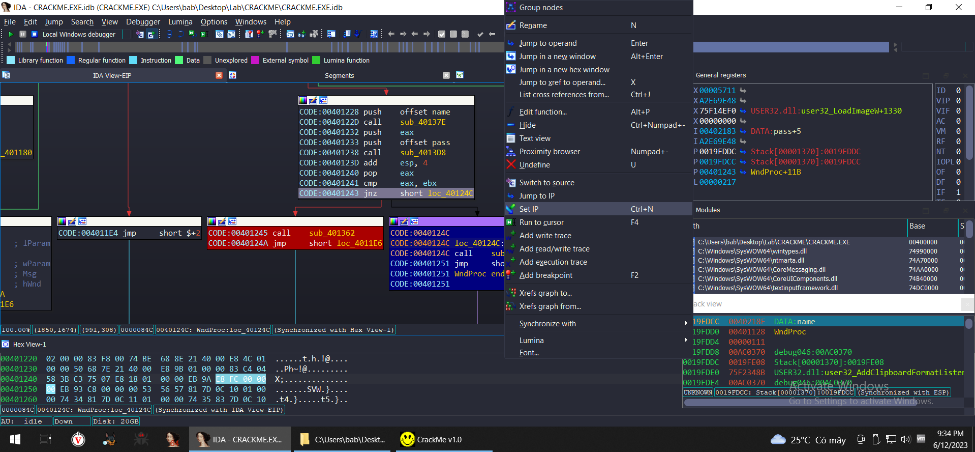
Đoạn code này cũng có một biến edi nhằm tính tổng của các ký tự trong password nhưng lần này phức tạp hơn nhiều. Từng ký tự trong password trước tiên sẽ được trừ đi cho 0x30, thanh ghi giữ tổng sẽ được nhân cho 0xA ở mỗi vòng lặp trước khi thực hiện phép cộng ký tự vào. Kết quả trong edi sau đó sẽ được đem đi xor với 0x1234 trước khi trả về trong thanh ghi ebx.

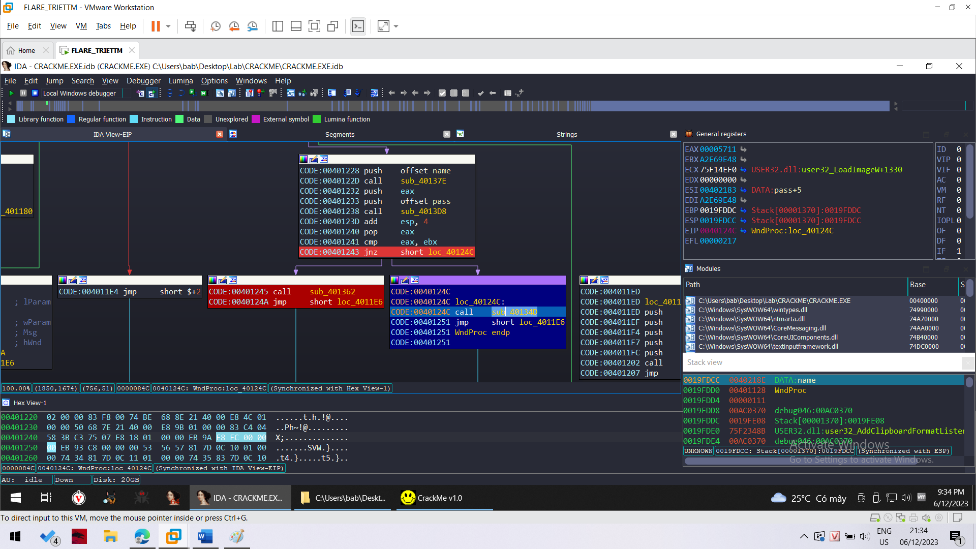


Sau đó 2 giá trị tổng ký tự của name và pass sẽ được đem đi so sánh với nhau. Nếu bằng thì đúng, không bằng là sai.

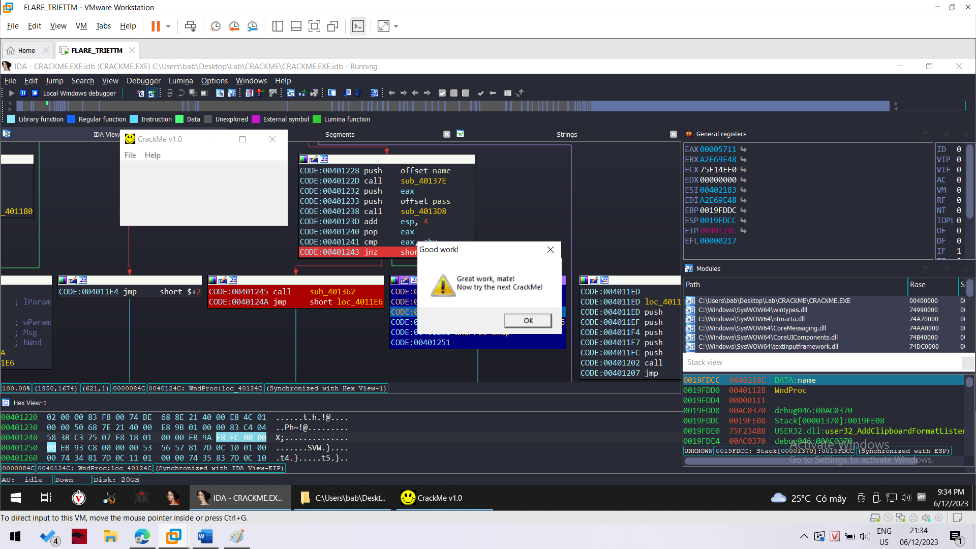
Ở đây sẽ có một cách làm dễ hơn là điều khiển luồng thực thi chương trình thay vì cố tìm cách tìm ra name và pass đúng đó là.



Cho chương trình chạy tới lệnh jump trước khi rẽ nhánh. Ta có thể chọn set IP để điều khiển con trỏ lệnh tới lệnh mà mình muốn cho chương trình thực thi tiếp. Ta chuyển vị trí con trỏ lệnh tới khing code màu xanh. 



Rồi cứ thế cho chương trình chạy tiếp thôi.



Dòng chữ greatwork đã hiện ra, vậy là ta đã thực hiện thành công crackme này rồi.