**Escuela Colombiana De Ingeniería**

**Julio Garavito**

**Seguridad y privacidad TI**

**Daniel Esteban Vela Lopez**

**Andres Felipe Montes**

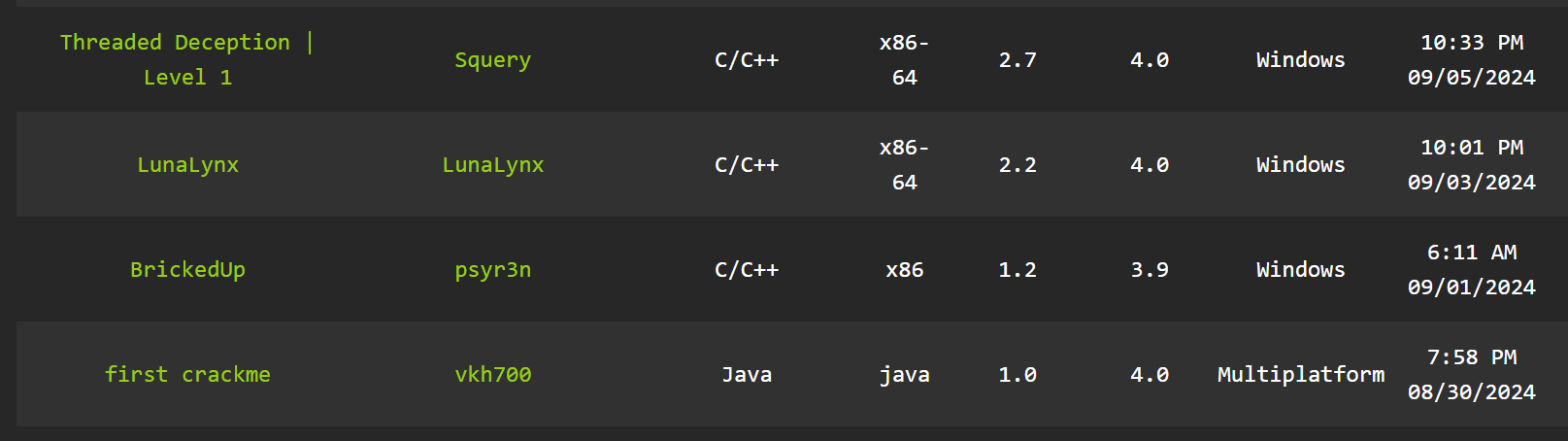
**Laura Valentina Rodriguez Ortegon**

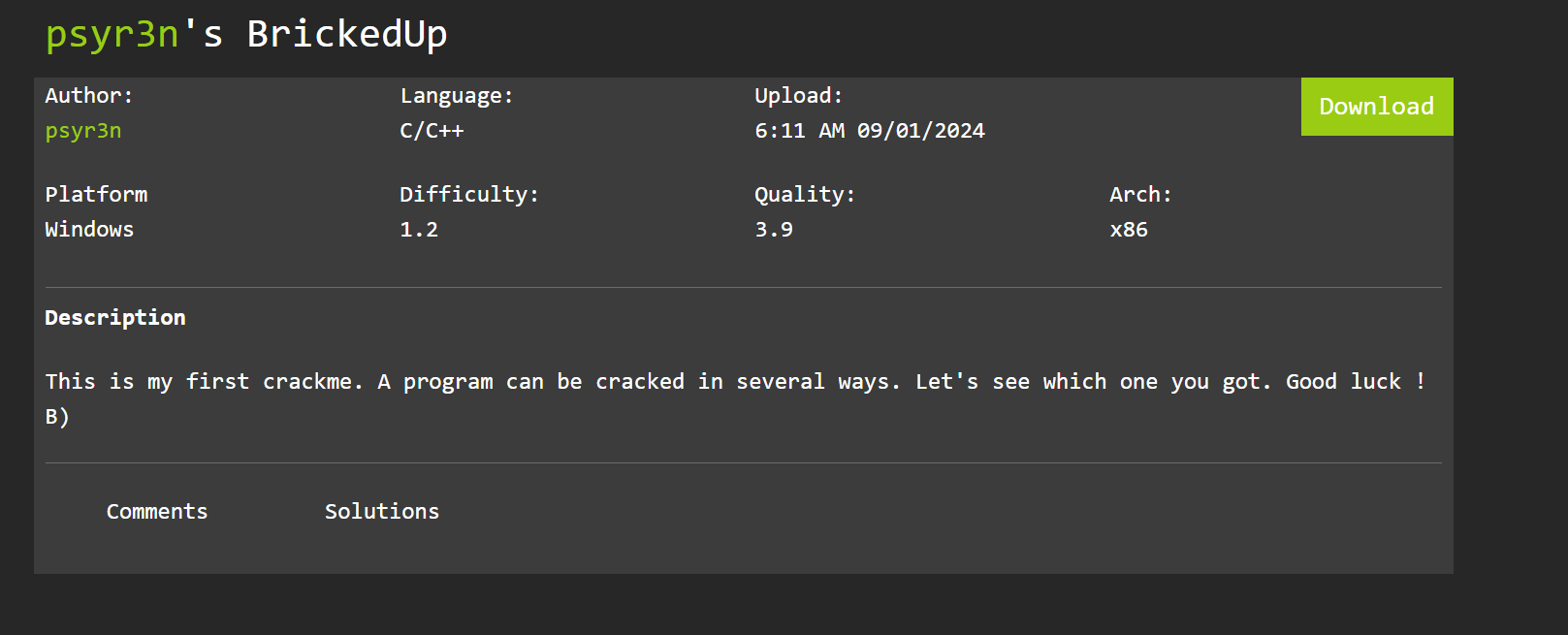
**Laboratorio No.6**

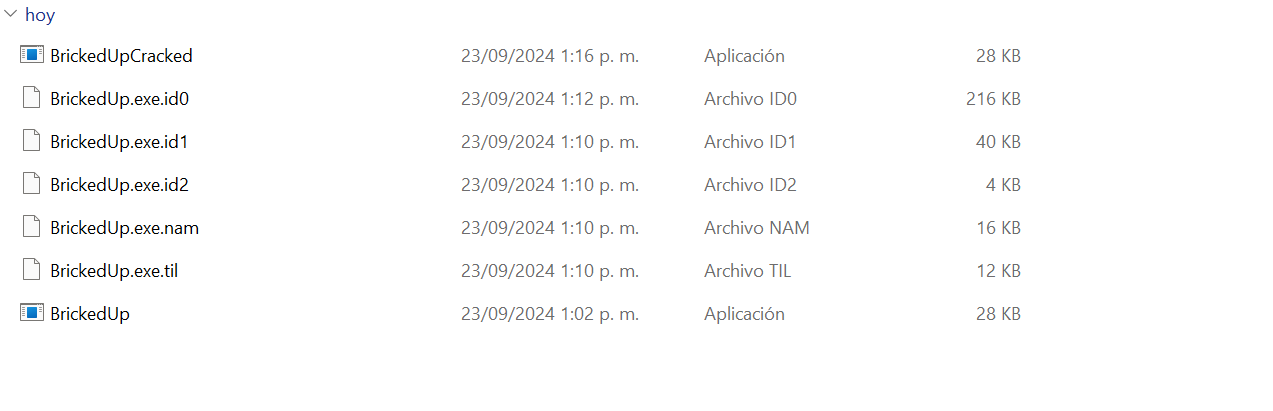
**2024-2**

**Primera: BrickedUp**

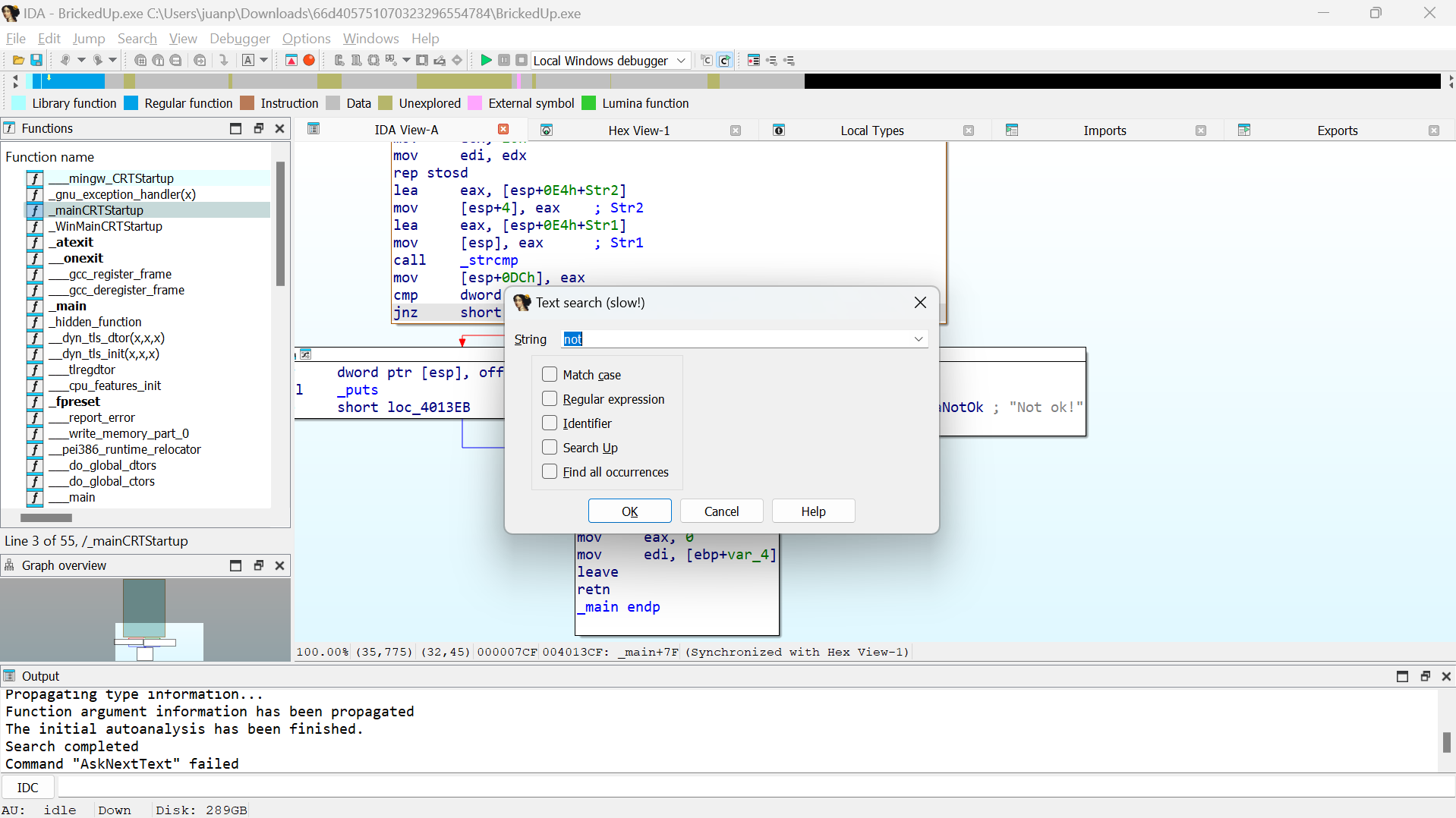
para empezar elegimos este programa en lenguaje c/c++ y con dificultad de 1.2

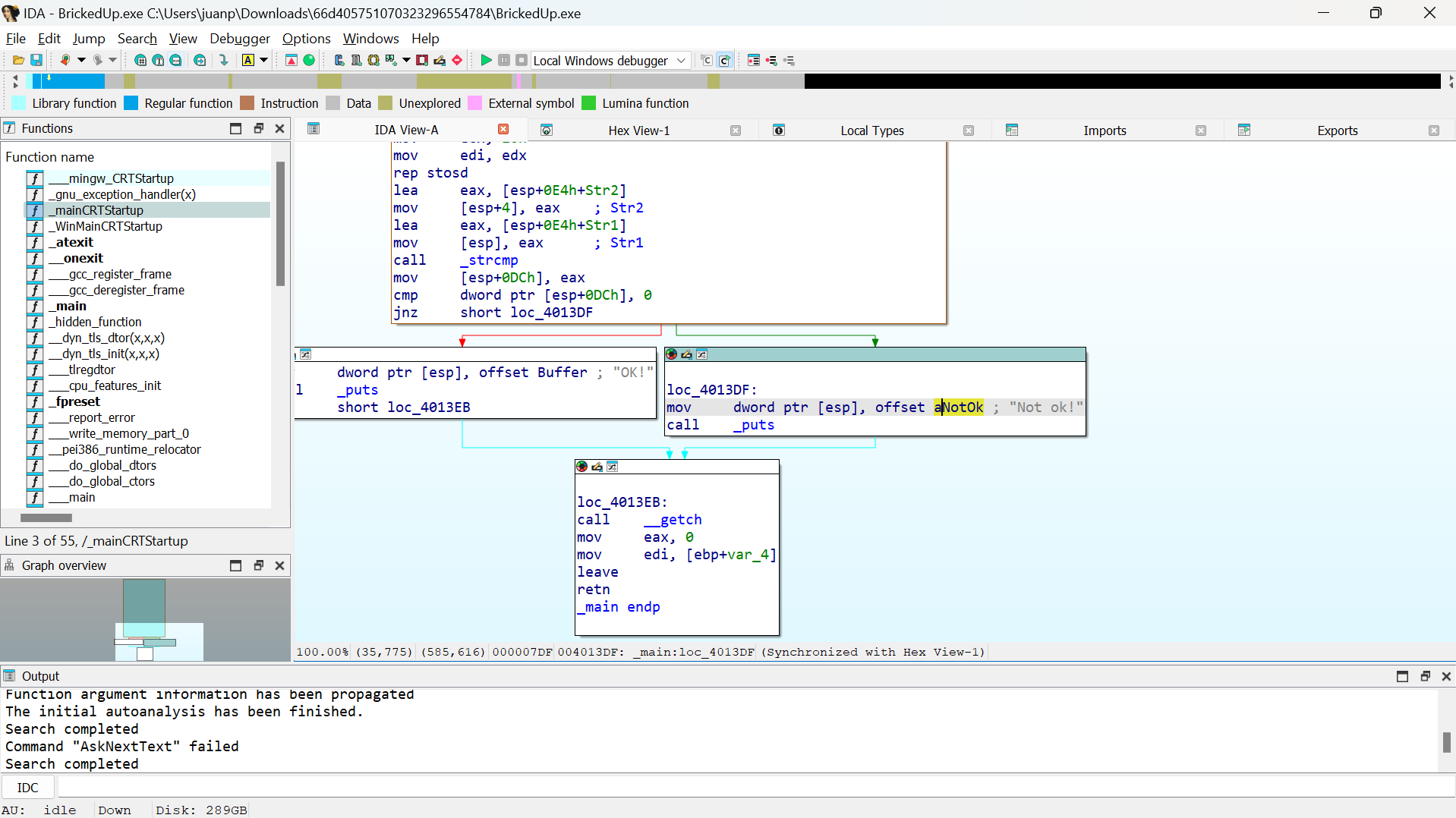




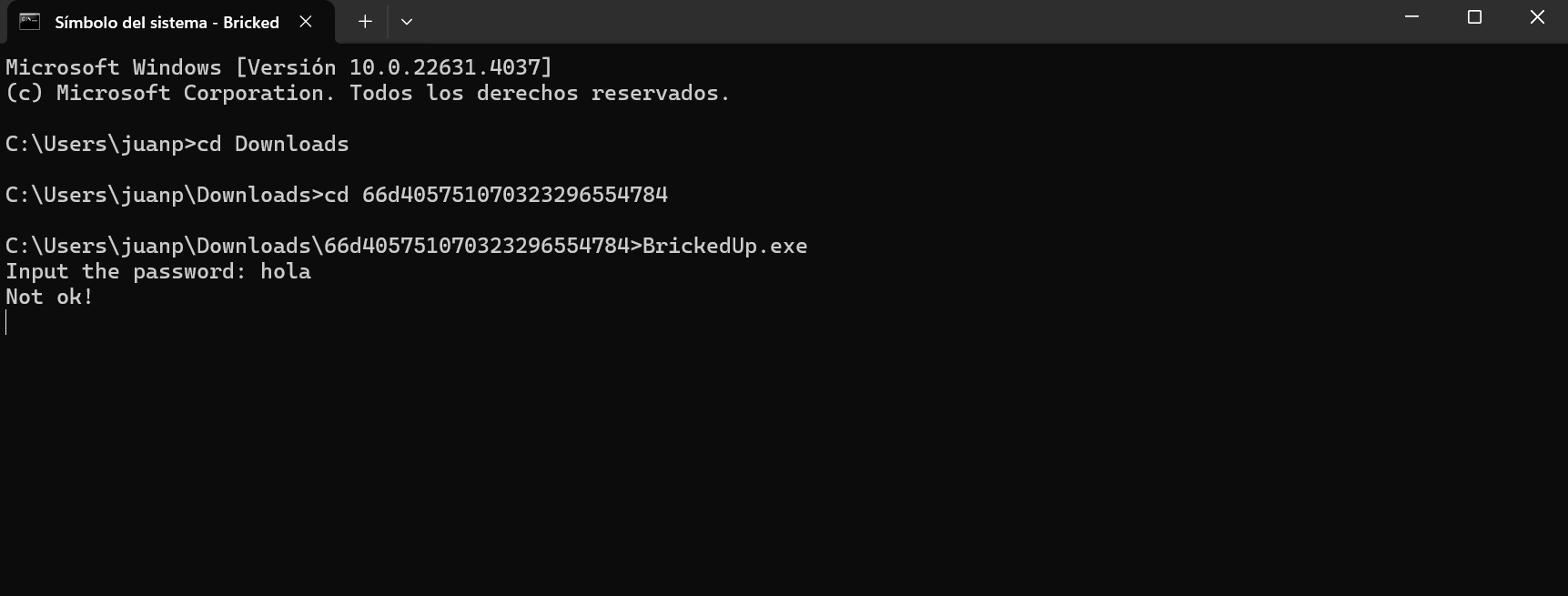


Después mediante la opción de búsqueda de “ida” ubicamos una posible entrada a la parte donde se verifica la respuesta correcta y podemos notar que se está haciendo una comparación.

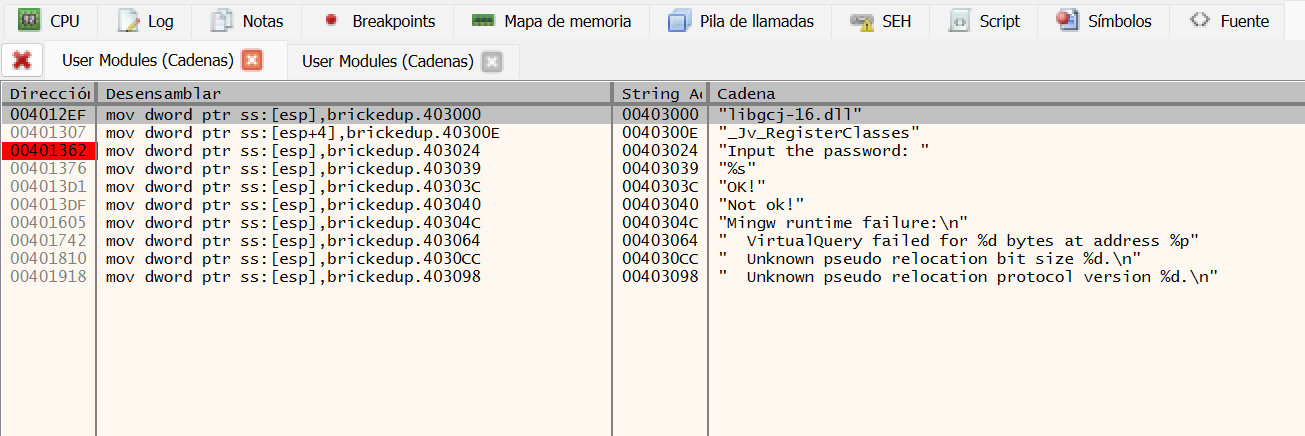




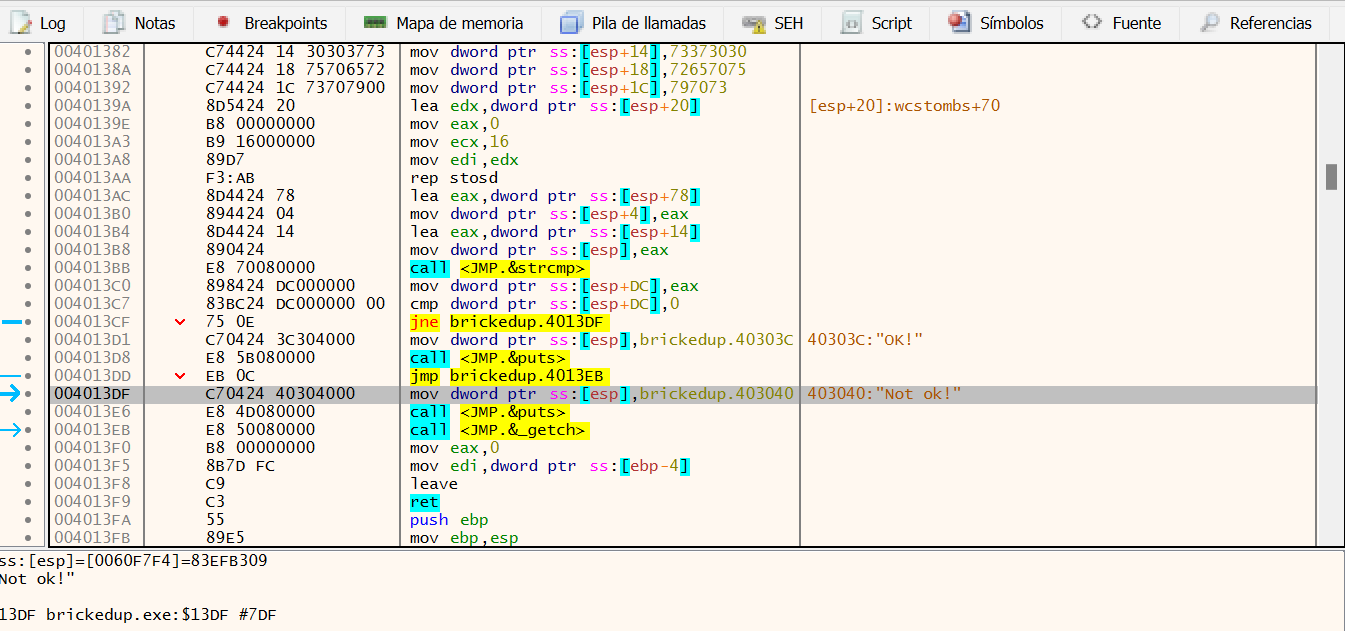
Entonces intentamos probar la entrada al programa con una contraseña cualquiera solo para verificar que no funciona y nos de un not ok!.



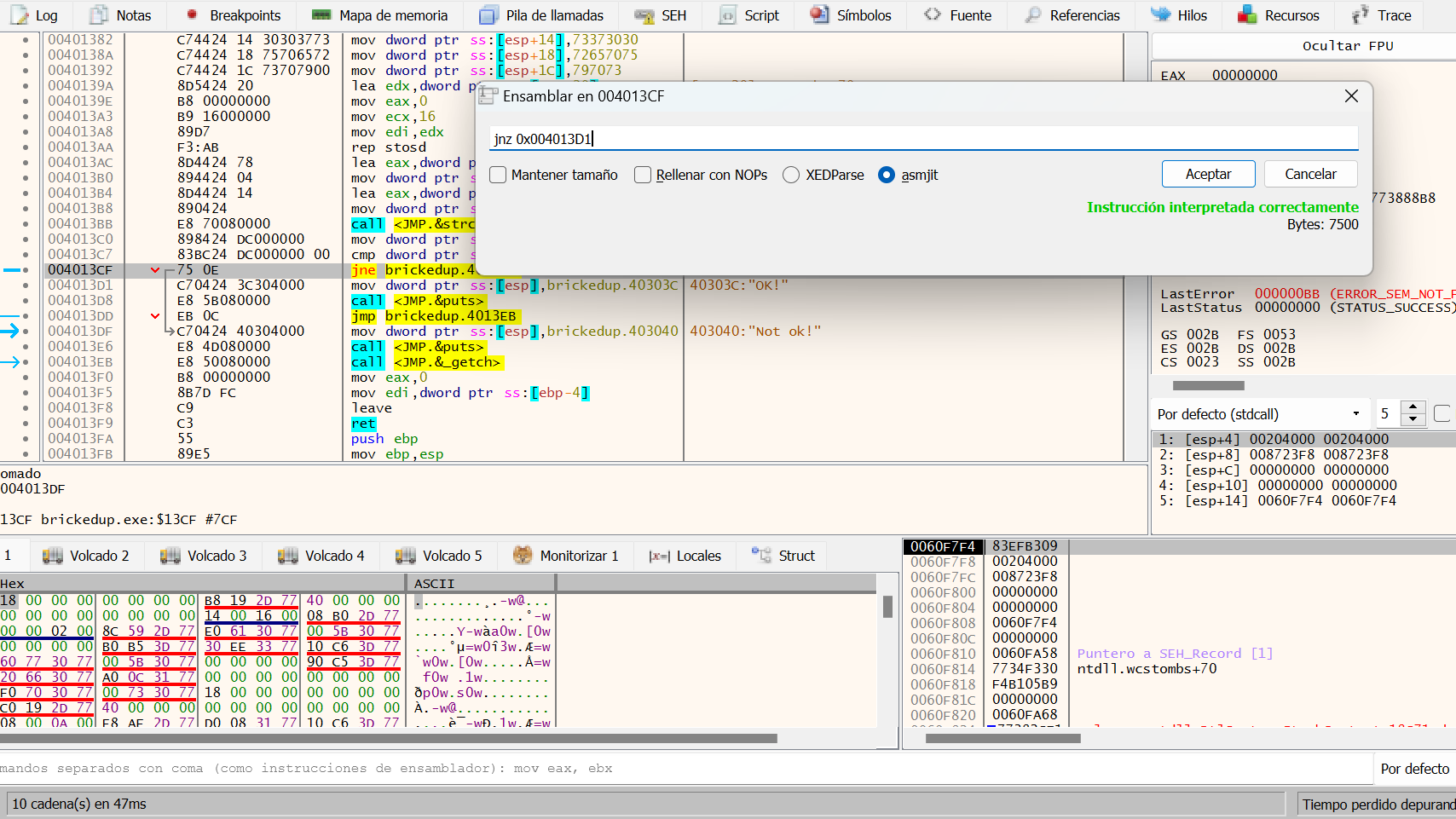
Abrimos el programa en x96dbg y en este caso seleccionamos x32, para posteriormente ir al filtrado por cadenas de texto: buscar en -> all user modules -> referencias de cadena.



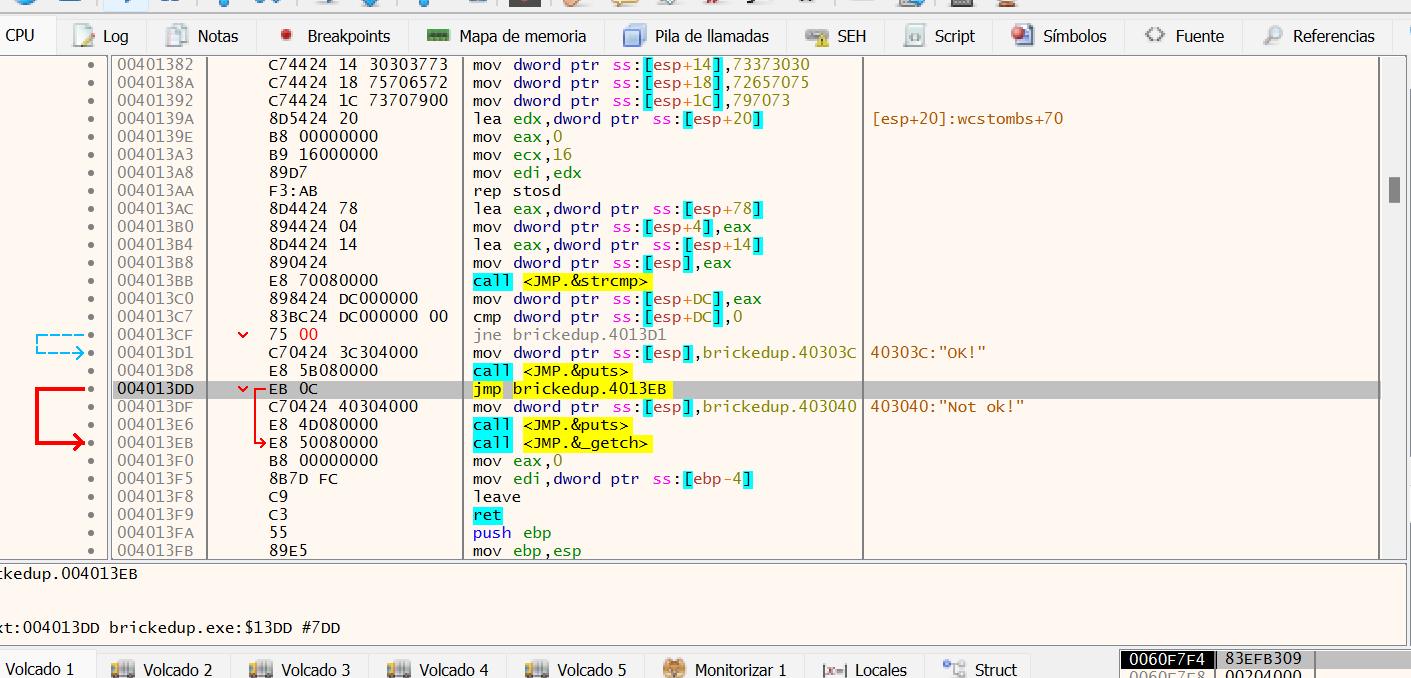
Ya con esto podemos ver que se hace una comparación y si esta es incorrecta hay un jump hasta la línea que nos muestra el not ok!.



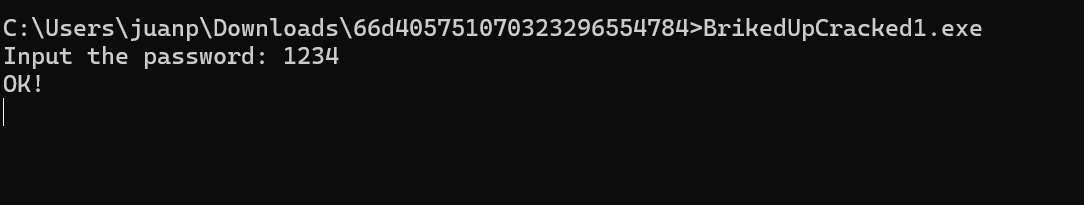
Entonces para poder hacer el crackeo modificamos la linea donde hace el jump para que siempre se dirija a “ok” .



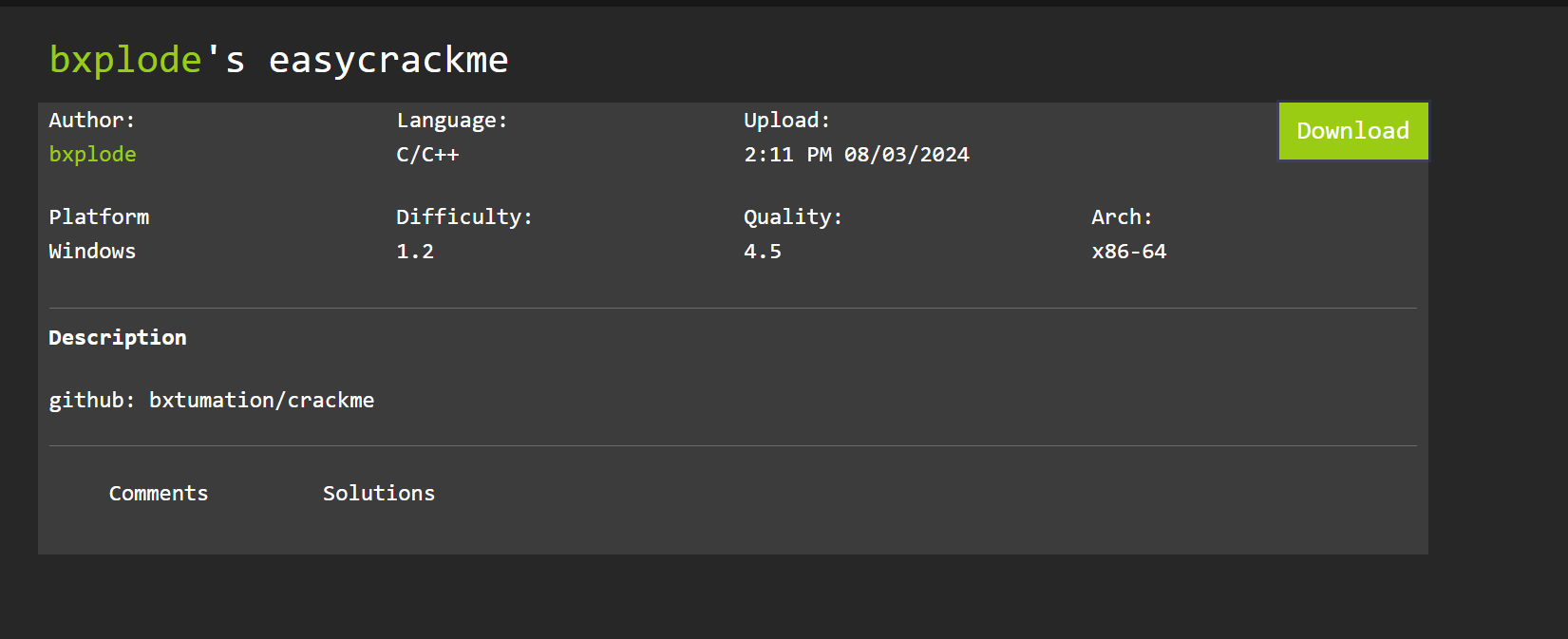
Aqui podemos apreciar como ahora siempre se dirije a la linea 004013d1.



Y por ultimo con una contraseña cualquiera podemos ingresar.



**Primera: easycrackme**



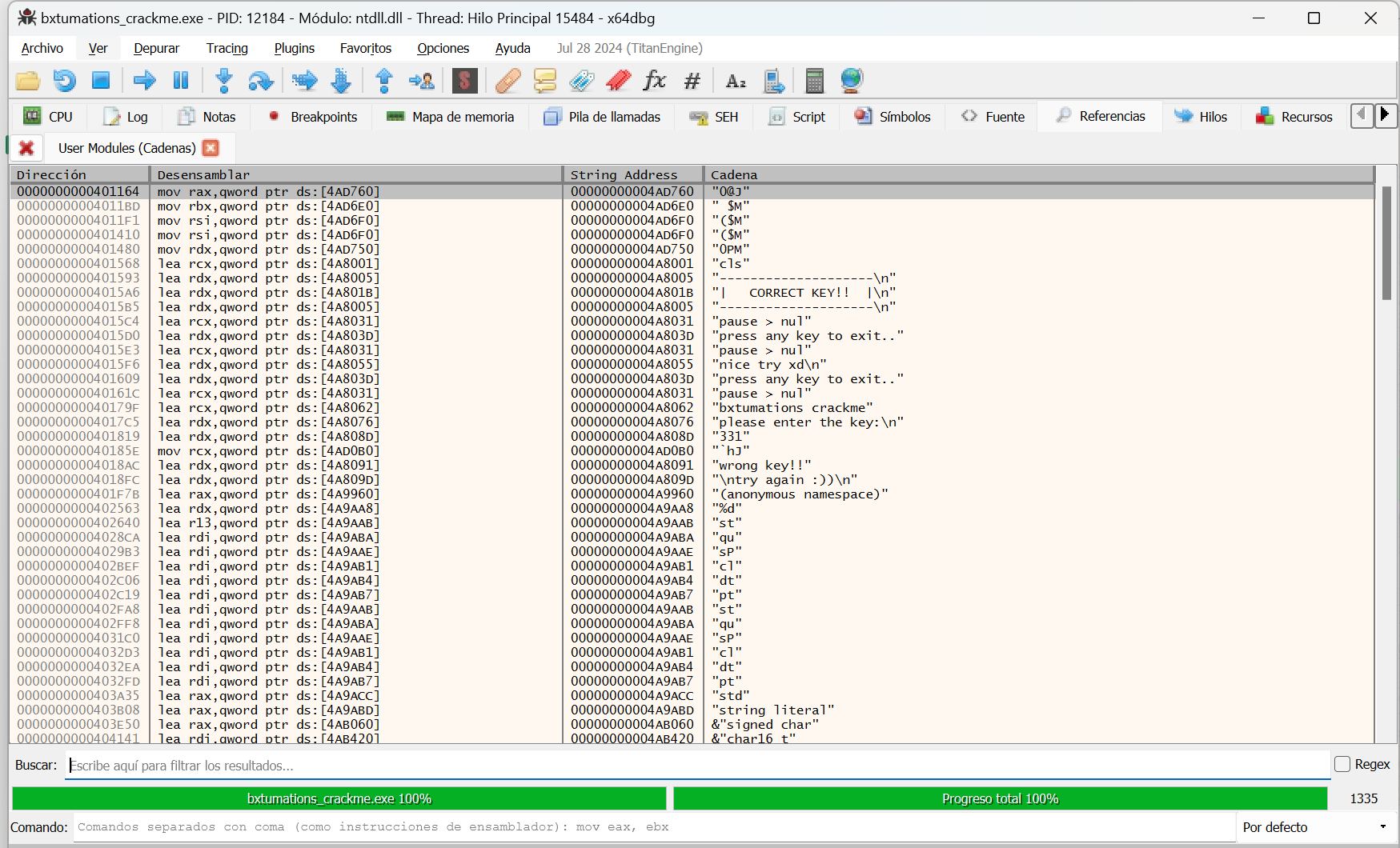
Para este programa podemos ver que el mensaje que muestra cuando ponemos una contraseña incorrecta es diferente “wrong key”.



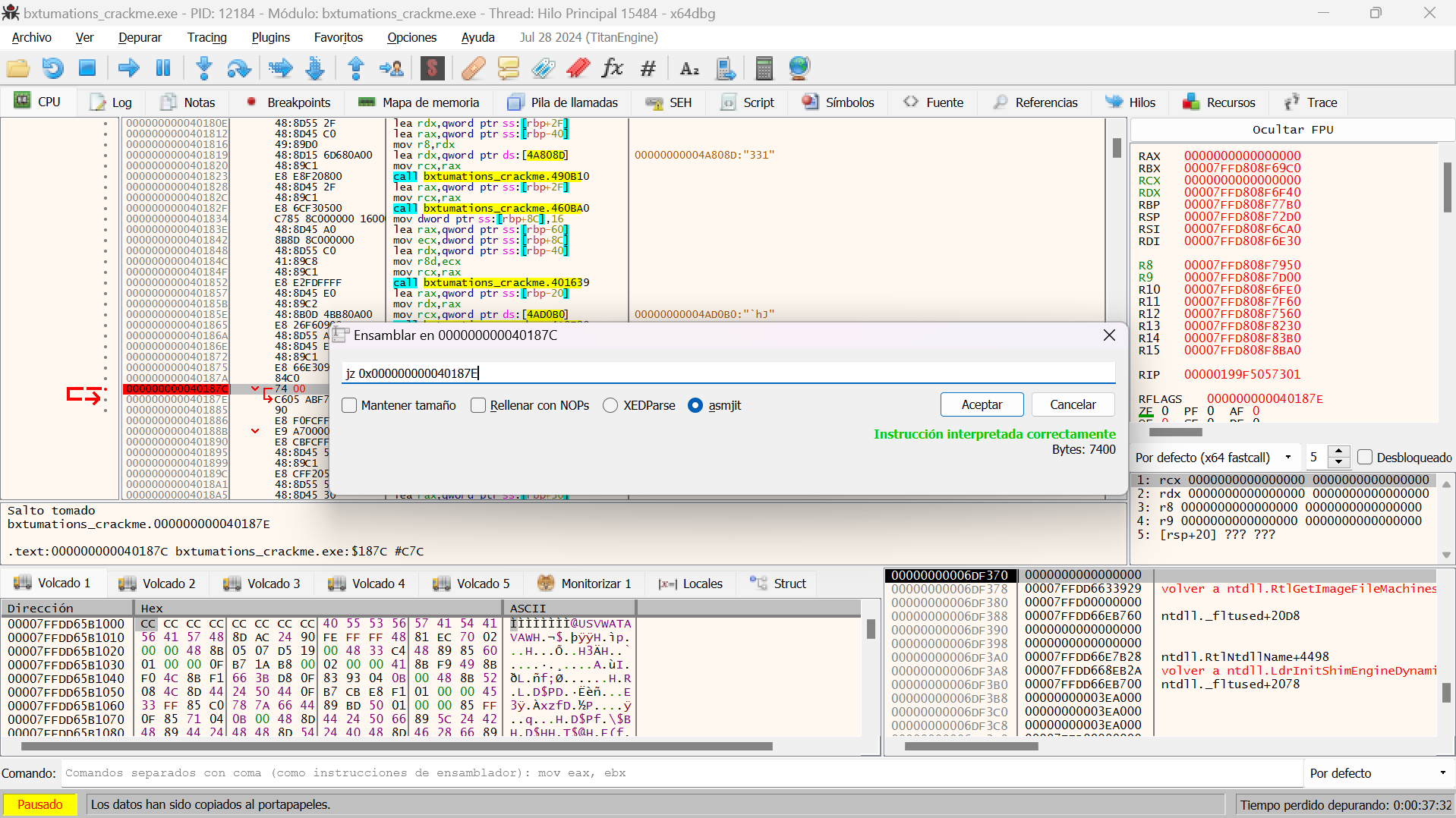
Entonces como lo hicimos en el caso anterior mediante ida y su filtrado de texto buscamos “wrong key” o “try again” para ubicarnos y ver si se hace alguna comparación.



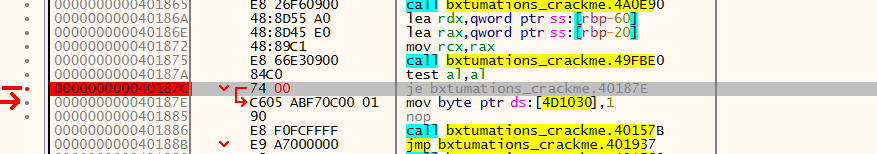
Después de eso abrimos el progama en x96dbg y en este caso usamos x64, como en el caso anterior filtramos, pero esta vez por cada una de las palabras clave “wrong key” y “Correct key”.



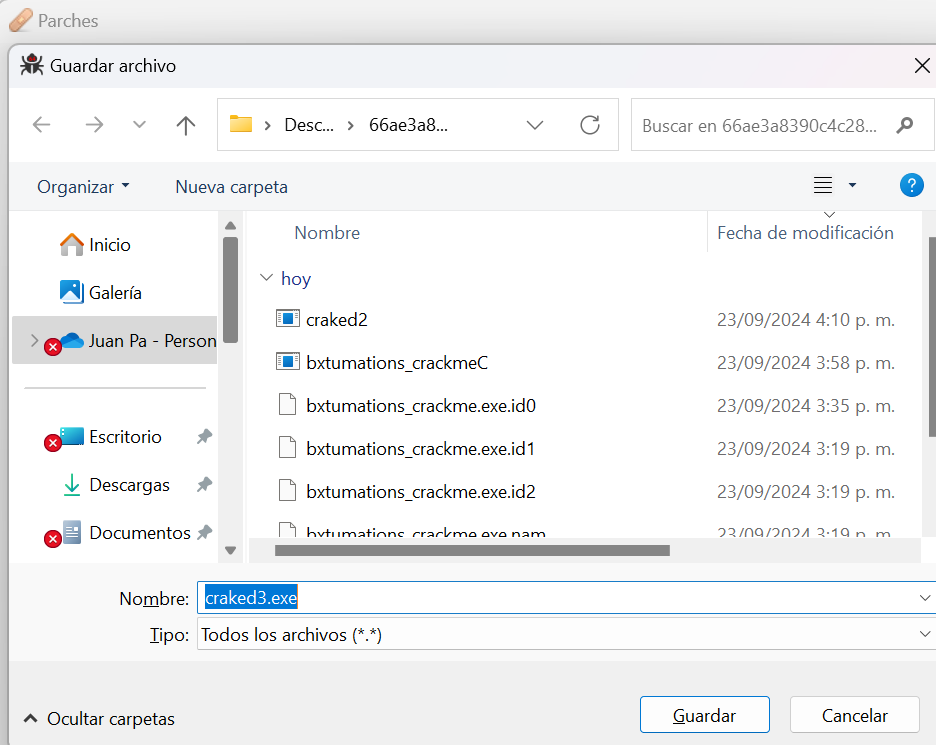
Después de navegar un rato y entender cómo es el flujo de información, podemos observar una “función” que dependiendo el valor 0/1 redirige a “wrong key” o “correct key”

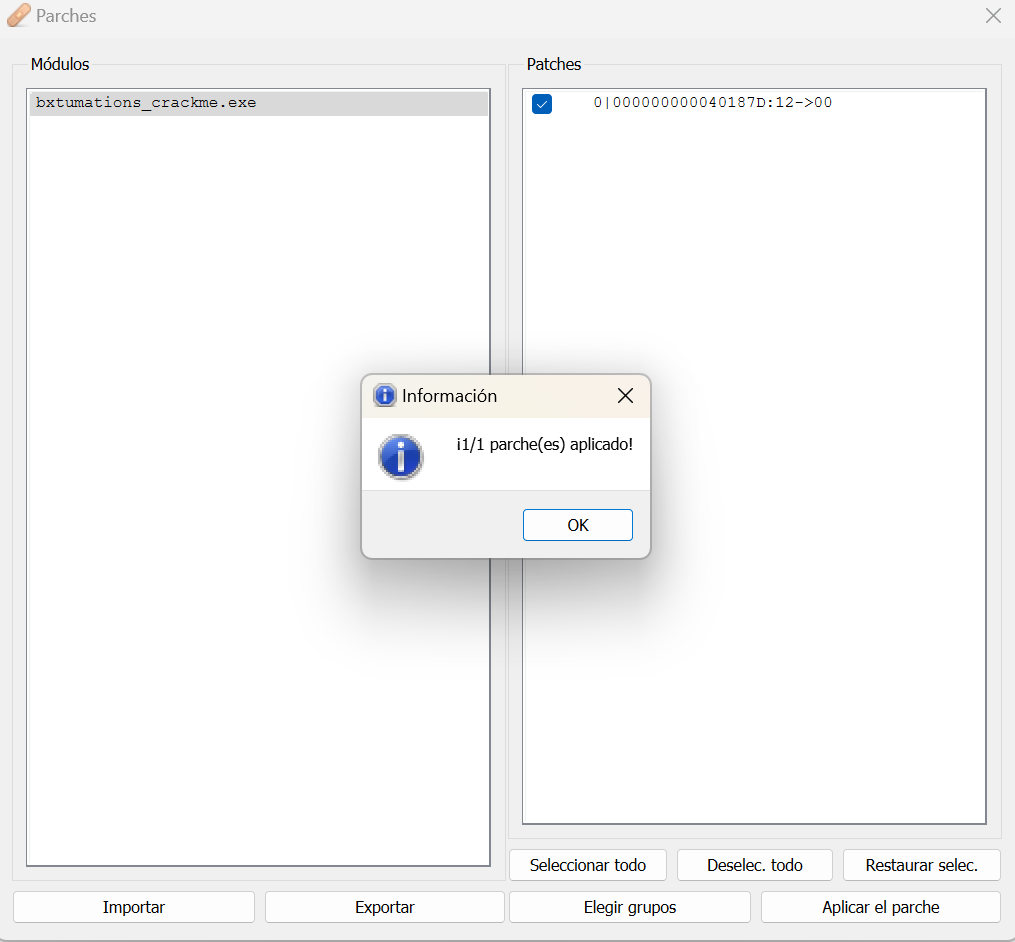


Entonces modificamos para que siempre se diría a 1 y por ende se dirija siempre a “correct key”.

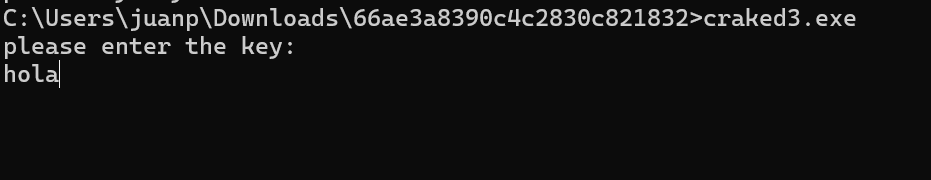


Guardamos el nuevo archivo crackeado.





Probamos que con una clase cualquiera podemos ingresar.



Yeii se logro :3



fin.