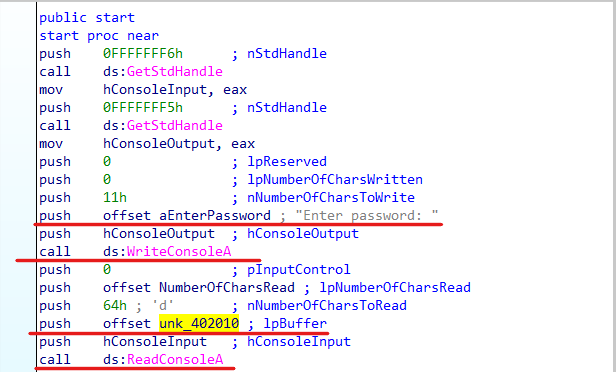
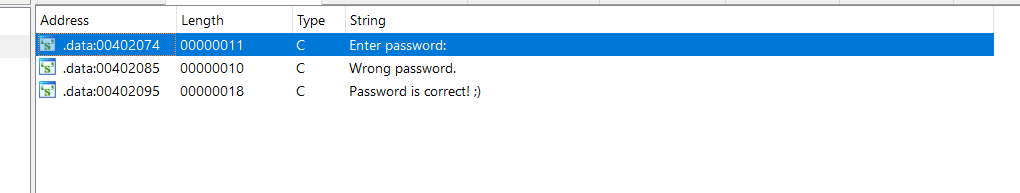
**Bài get\_the\_password**

****Nhìn sơ vào hàm main ta thấy chương trình thực hiện in ra dòng “Enter password: “ và đọc input người dùng nhập vào lưu ở địa chỉ unk\_402010.

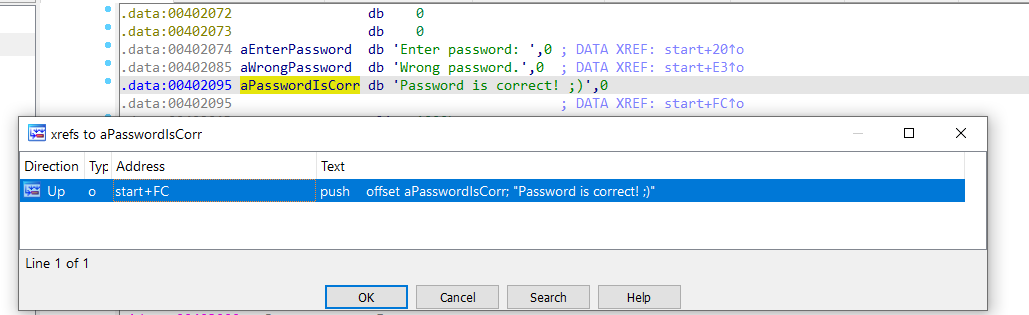
Bấm Shift F12 để xem string trong chương trình (công việc quen thuộc):

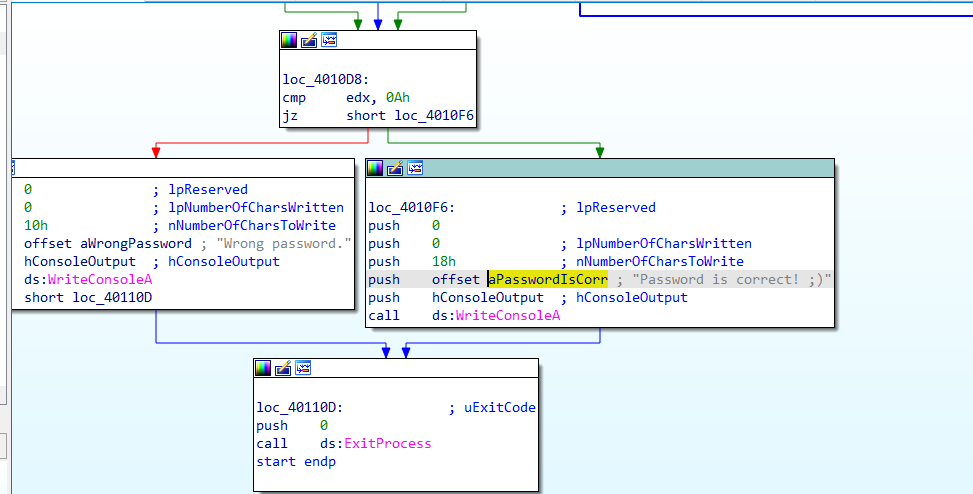


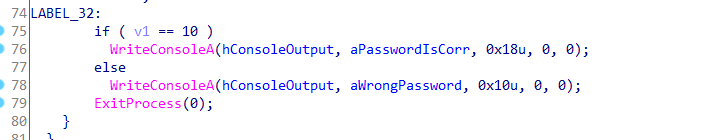
Nhấn đúp vào string mong muốn Password is correct! ; )



Bấm X tiếp tục để xem các nơi có đề cập đến string này trong chương trình, chọn rồi ok.

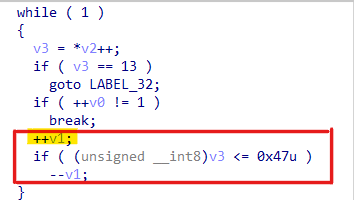


Bấm F5 để xem mã giả.

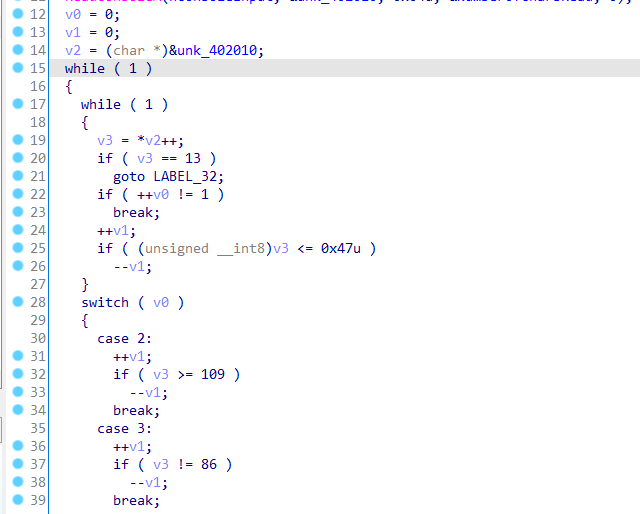
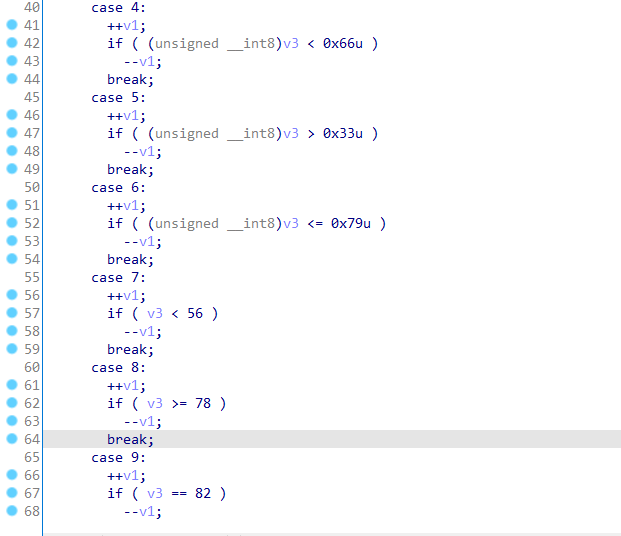
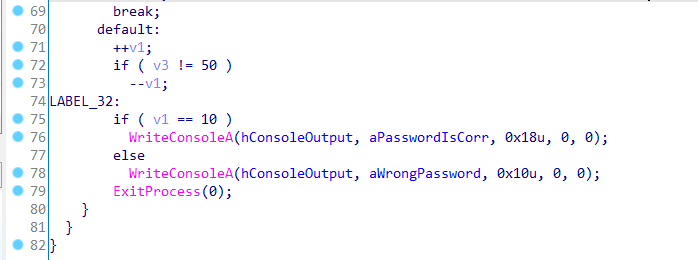
Xem mã ta thấy ở cuối có điều kiện so sánh nếu v1=10 thì in ra aPasswordIsCorr, ngược lại in ra aWrongPassword. Vì vậy ta sẽ tìm cách để v1=10.

Ban đầu v0=0, v1=0, v2 trỏ đến địa chỉ unk\_402010( input người dùng nhập vào lưu ở địa chỉ này).Ta thấy vòng lặp while nhỏ (lồng bên trong vòng lặp while lớn) thực hiện: gán v3 là giá trị tại v2, tăng v2++, tăng v0 thành 1, tăng v1 thêm 1, so sánh kí tự đầu tiên tại v3 (kí tự đầu của chuỗi người dùng nhập vào) nếu <=0x47 thì - -v1.

Do giá trị v0 không bị thay đổi trong switch nên đoạn code sau đây chỉ được thực hiện 1 lần (lần lặp đầu tiên).



Để ý thấy v0 tăng dần 1 đơn vị trong vòng lặp while (tương ứng xét trường hợp từng từ kí tự trong chuỗi nhập vào) và vòng lặp while lớn thoát khi v0=10. Từ case v0=1(đoạn code khoanh đỏ trên), case v0=2 đến v0=10(default) trong đoạn code đều có ++v1 và (- -v1 nếu điều kiện if đúng). Vì vậy ý tưởng là ta có thể nhập input sao cho từng kí tự không thỏa điều kiện if tương ứng thế thì sẽ có 10 lần ++v1, cuối cùng v1=10 in ra correct password.



Vậy điều kiện cho từng kí tự sẽ là:

+ v0=1, v3 <= 0x47

+ v0=2, v3 <109

+ v0=3, v3 = 86

+ v0=4, v3 >= 0x66

+ v0=5, v3 <= 0x33

+ v0=6, v3 > 0x79

+ v0=7, v3 >=56

+ v0=8, v3 <78

+ v0=9, v3 !=82

+ v0=10, v3 =50

Có rất nhiều nghiệm. Chọn một nghiệm thỏa chẳng hạn HlVf3z8MQ2

Kiểm tra:

