**LẬP TRÌNH HỆ THỐNG  
BÁO CÁO LAB 4**

**KỸ THUẬT DỊCH NGƯỢC CƠ BẢN**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Lớp** |
| Hồ Diệp Huy | 22520541 | NT209.O21.ANTT.2  Nhóm 14 |
| Mai Nguyễn Nam Phương | 22521164 |
| Đinh Quốc Huy | 22520536 |

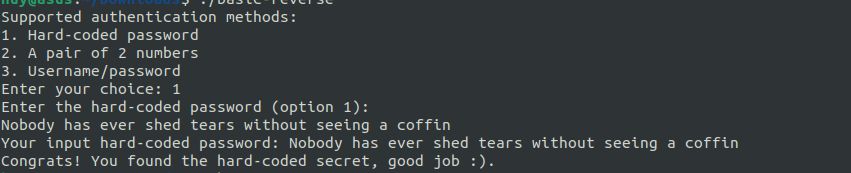
A close-up of a white background

Description automatically generated

Hình 1: Kết quả trả lời câu hỏi

# C2.1:

**Kết quả:**



Hình 2: Kết quả khi chạy Câu 1

**Giải thích:** Dựa vào hàm call \_isoc99\_scanf ( màu đỏ ) để lấy giá trị đầu vào, sau đó chương trình sẽ push kết quả đúng và input của ta vào stack rồi so sánh kết quả xem 2 chuỗi kí tự có tương tự nhau không nhờ call hàm \_strcmp ( màu tím ), nếu 2 chuỗi là tương tự thì ta sẽ thành công ( màu đen ) (Dễ dàng nhận thấy chuỗi kí tự đúng sẽ là s2)

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

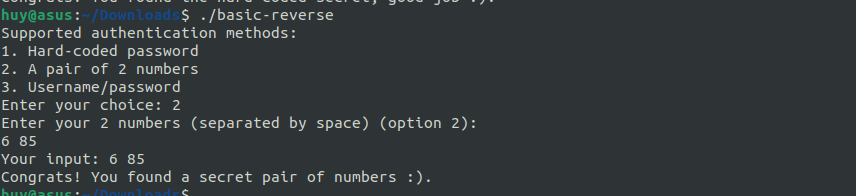
Mô tả được tạo tự động

Chuỗi kí tự đúng

Hình 3: Hàm hardCode của Câu 1

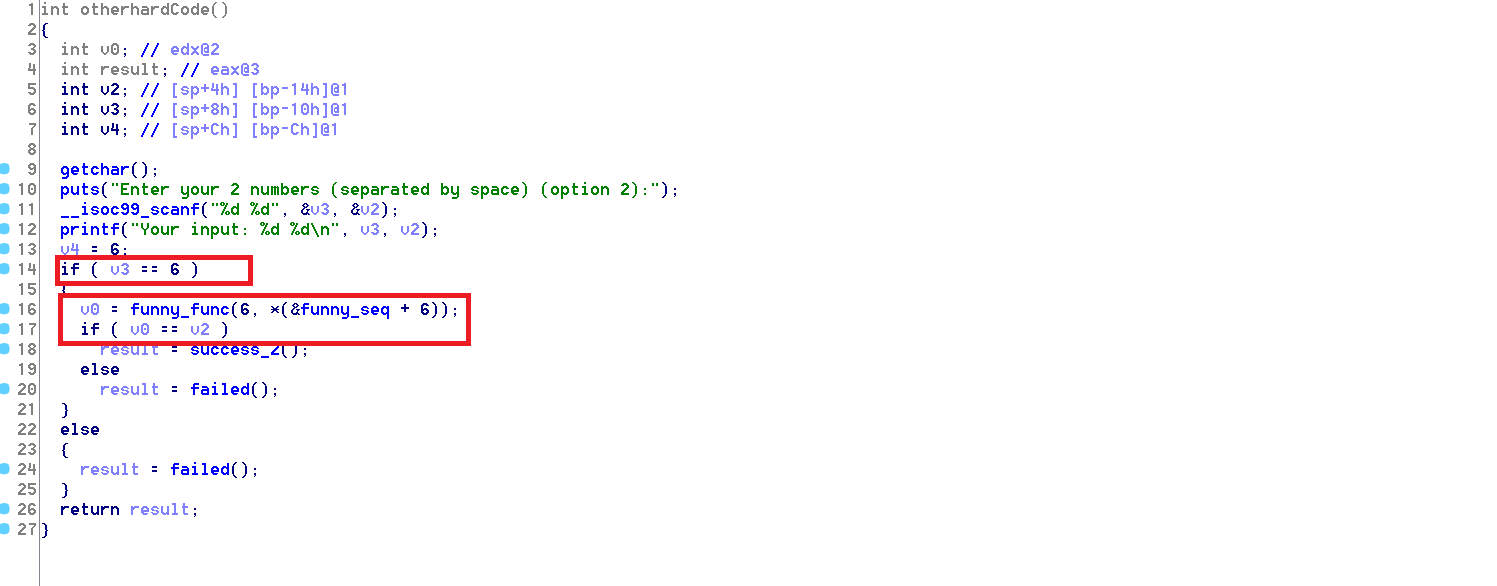
# C2.2:

**Kết quả:**



Hình 4: Kết quả khi chạy Câu 2

**Giải thích:**



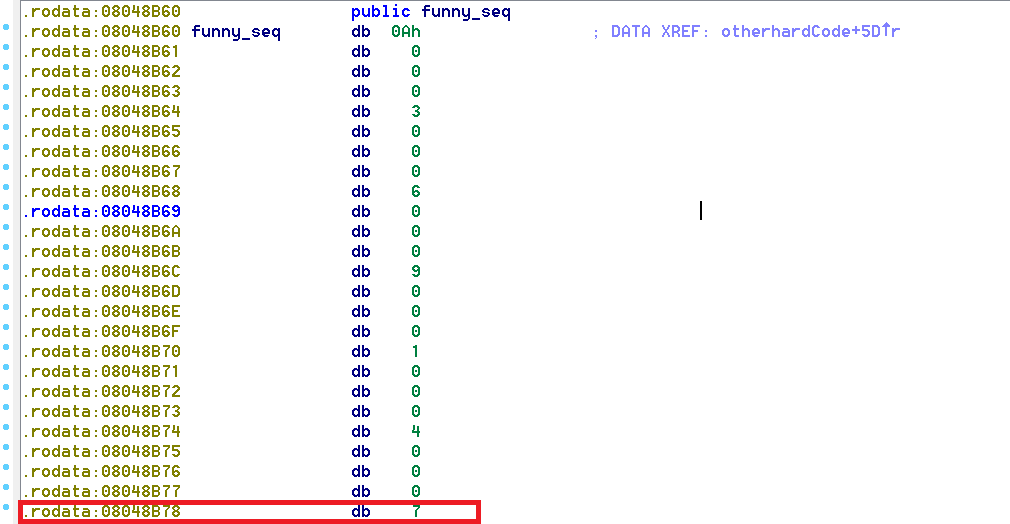
Hình 5: Hàm otherhardCode của Câu 2

- Ở lệnh if đang so sánh nếu v3 == 6 và v0 == v2 thì gọi hàm success\_2() nên v3 và v0 là 2 số cần tìm.

- Lệnh if đầu tiên giả sử **v3 = 6** nên số đầu tiên là **6**, sau đó xét tiếp số thứ 2.

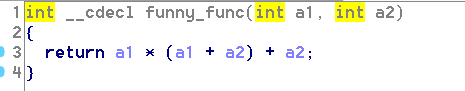
- Chương trình gán **v0 = funny\_func(6, \*(&funny\_seq + 6))**

- Tìm đến **funny\_seq** ta lấy kết quả của ô có địa chỉ là (&funny\_seq + 6).



Hình 6: Stack funny\_seq

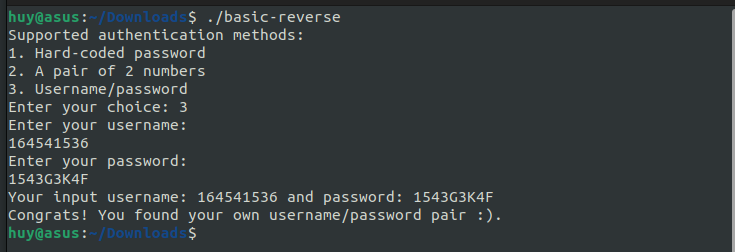
- Tìm đến hàm **funny\_func()** ta thấy hàm **funny\_func(6, \*(&funny\_seq + 6))** trả về kết quả như sau: 6 \* ( 6 + 7 ) + 7 = 85



Hình 7: Hàm funny\_func

**=> Vậy hai số cần tìm là 6 và 85.**

# C2.3:

**Kết quả:** 

**Giải thích:**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

- Dựa trên đoạn mã giả thì ta có thể chuyển về code C/C++ để dễ tìm ra password thông qua username hơn:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

- Kết quả của đoạn code:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

**-** Phân tích:

* Đoạn code này nhằm giúp ta tìm ra được password cần nhờ vào username nhập vào
* Đầu tiên ta có khai báo một số biến cần thiết, bao gồm **v8** chứa một mảng các ký tự đã được khai báo sẵn (ở đây, các phần tử được khai báo trong mảng **v8** chính là mã ASCII), **s** chứa tên người dùng nhập vào và **v6** là mật khẩu cần tìm
* Sau đó tiến hành kiểm tra xem độ dài của tên người dùng có phù hợp không (9 ký tự).
* Nếu phù hợp thì sẽ tiếp tục lần lượt kiểm tra điều kiện của 1 biến i ( khởi tạo i = 0, cho i chạy dần tới 8) nhằm tìm ra lần lượt các kí tự của chuỗi kí tự v5
* Tiếp tục ta lại xét 1 biến i khác ( vẫn khởi tạo bằng 0 rồi cho chạy dần ) ở đây ta khởi tạo biến v3 = chiều dài của username ( = 9 ) rồi cho chạy vòng lặp so sánh dần dần với i, nếu v3 > i thì mới tiếp tục
* Thực hiện lấy các kí tự của chuỗi password nhờ lấy trung bình cộng của tổng mã ASCII giữa từng kí tự chuỗi s và chuỗi v5

**-** Thực hiện giải tay:

Dựa trên đề cho thì ta có mảng chứa username có 9 kí tự: 164541536, ta sẽ được mã ASCII như sau:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| s[i] | 1 | 6 | 4 | 5 | 4 | 1 | 5 | 3 | 6 |
| ASCII | 49 | 54 | 52 | 53 | 52 | 49 | 53 | 51 | 54 |

- Xét vòng lặp for đầu tiên: **for unsigned i = 0; i <= 8; ++i**, ta được giá trị của các phần tử mảng **v5** như sau (chú thích: hàng 3 là giá trị cụ thể của mỗi phần tử v5[i] bao gồm ký tự hiển thị và mã ASCII tương ứng)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| v5[i] | s[5] | s[6] | s[4] | s[5] | v8[4] | v8[3] | v8[2] | v8[1] | v8[0] |
| Giá trị v4[i] | ‘1’ 49 | ‘5’ 53 | ‘4’ 52 | ‘1’ 49 | ‘Z’ 90 | ‘6’ 54 | ‘b’ 98 | ‘6’ 54 | ‘W’ 87 |

- Xét vòng lặp for tiếp theo for **i = 0;i<9;++i** ta được phần tử của password như sau:

Lưu ý: các số thập phân ở đây làm tròn xuống

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 |
| v6[i] | (s[0]+v5[0])/2 = (49+49)/2 = 49 | (s[1]+v5[1])/2 = (53+54)/2 = 53 | (s[2]+v5[2])/2 = (52+52)/2 = 52 | (s[3]+v5[3])/2 = (53+49)/2 = 51 | (s[4]+v5[4])/2 = (90+52)/2 = 71 |
| Giá trị của v6[i] | ‘1’ | ‘5’ | ‘4’ | ‘3’ | ‘G’ |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 6 | 7 | 8 |
| (s[5]+v5[5])/2 = (49+54)/2 = 51 | (s[6]+v5[6])/2 = (53+98)/2 = 75 | (s[7]+v5[7])/2 = (51+54)/2 = 52 | (s[8]+v5[8])/2 = (87+54)/2 = 70 |
| ‘3’ | ‘K’ | ‘4’ | ‘F’ |

Từ bảng trên ta có thể có được password là 1543G3K4F