Lab 3

Student 1: Võ Anh Kiệt

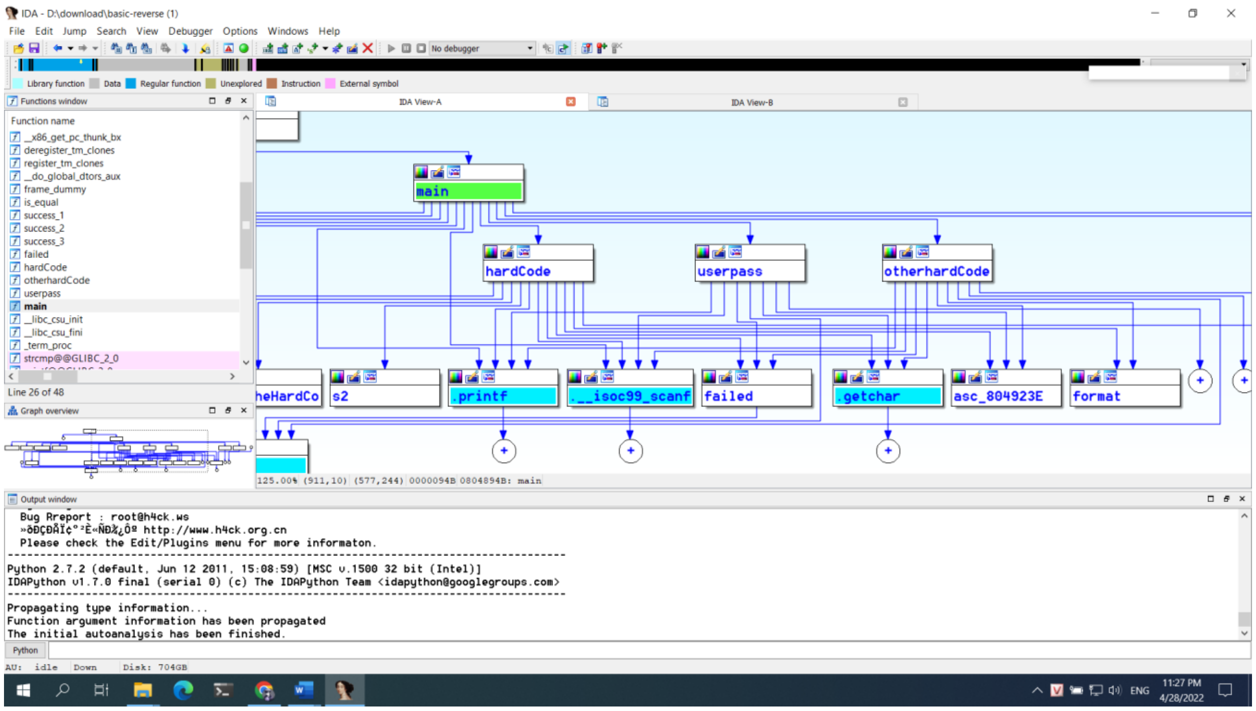
ID Student 1: 20520605

Student 2: Nguyễn Bảo Phương

ID Student 2: 20520704

Class: NT209.M21.ANTN

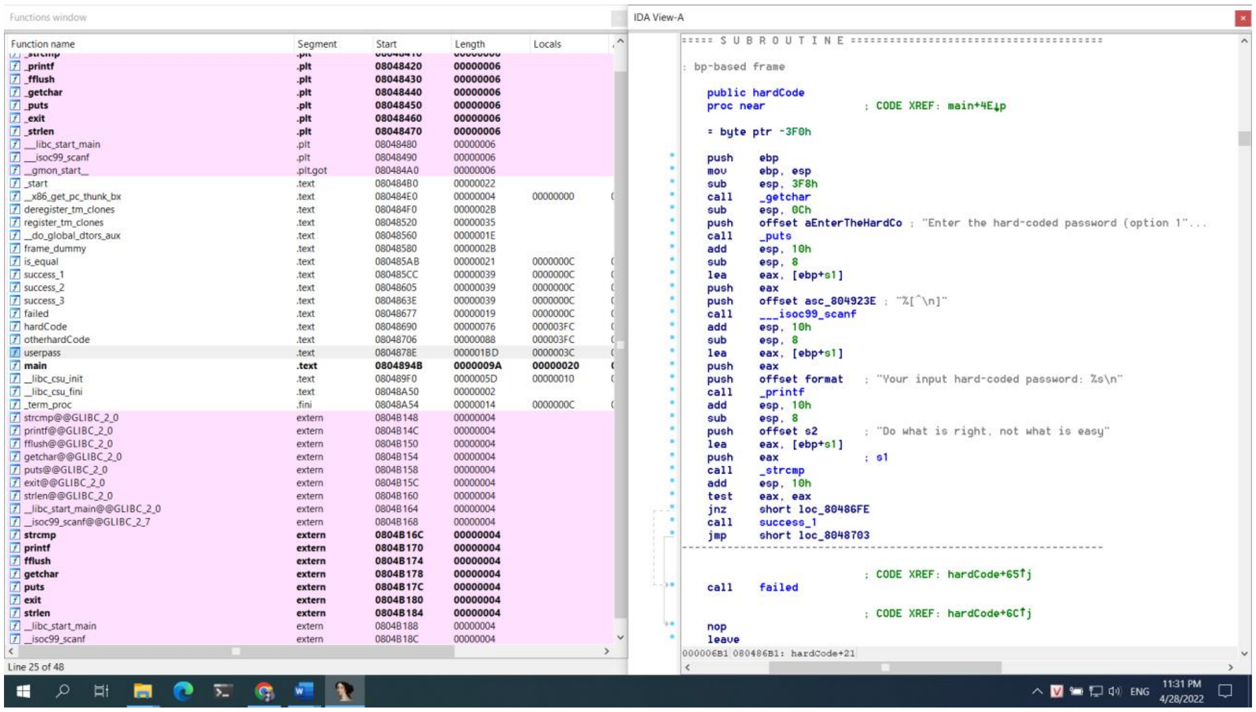
It is easy to see that the main function has 3 tasks: hardCore, otherhardCore and userpass



```
  Bug Rreport : root@h4ck.ws
  »ðĐÇĐÅÏ¢º²È«ÑĐ¾¿Ôº http://www.h4ck.org.cn
  Please check the Edit/Plugins menu for more informaton.
---------------------------------------------------------------------------------------
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)]
IDAPython v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
---------------------------------------------------------------------------------------
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

Task 1: Do what is right, not what is easy

Check the function for the task 1. The function for the task 1 is hardCore



Functions window

Function name	Segment	Start	Length	Locals
_printf	.plt	08048420	00000006	
_fflush	.plt	08048430	00000006	
_getchar	.plt	08048440	00000006	
_puts	.plt	08048450	00000006	
_exit	.plt	08048460	00000006	
_strlen	.plt	08048470	00000006	
___libc_start_main	.plt	08048480	00000006	
___isoc99_scanf	.plt	08048490	00000006	
__gmon_start__	.plt.got	080484A0	00000006	
_start	.text	080484B0	00000022	
__x86_get_pc_thunk_bx	.text	080484E0	00000004	00000000
deregister_tm_clones	.text	080484F0	0000002B	
register_tm_clones	.text	08048520	00000035	
__do_global_dtors_aux	.text	08048560	0000001E	
frame_dummy	.text	08048580	0000002B	
is_equal	.text	080485AB	00000021	0000000C
success_1	.text	080485CC	00000039	0000000C
success_2	.text	08048605	00000039	0000000C
success_3	.text	0804863E	00000039	0000000C
failed	.text	08048677	00000019	0000000C
hardCode	.text	08048690	00000076	000003FC
otherhardCode	.text	08048706	00000088	000003FC
userpass	.text	0804878E	000001BD	0000003C
main	.text	0804894B	0000009A	00000020
__libc_csu_init	.text	080489F0	0000005D	00000010
__libc_csu_fini	.text	08048A50	00000002	
_term_proc	.fini	08048A54	00000014	0000000C
strcmp@@GLIBC_2_0	extern	0804B148	00000004	
printf@@GLIBC_2_0	extern	0804B14C	00000004	
fflush@@GLIBC_2_0	extern	0804B150	00000004	
getchar@@GLIBC_2_0	extern	0804B154	00000004	
puts@@GLIBC_2_0	extern	0804B158	00000004	
exit@@GLIBC_2_0	extern	0804B15C	00000004	
strlen@@GLIBC_2_0	extern	0804B160	00000004	
__libc_start_main@@GLIBC_2_0	extern	0804B164	00000004	
__isoc99_scanf@@GLIBC_2_7	extern	0804B168	00000004	
strcmp	extern	0804B16C	00000004	
printf	extern	0804B170	00000004	
fflush	extern	0804B174	00000004	
getchar	extern	0804B178	00000004	
puts	extern	0804B17C	00000004	
exit	extern	0804B180	00000004	
strlen	extern	0804B184	00000004	
__libc_start_main	extern	0804B188	00000004	
__isoc99_scanf	extern	0804B18C	00000004	

Line 25 of 48

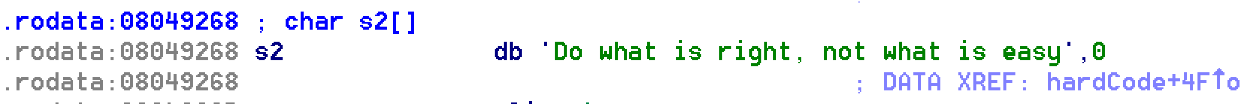
IDA View-A

```
===== S U B R O U T I N E =======================================

; bp-based frame

  public hardCode
  proc near                ; CODE XREF: main+4E↓p

  = byte ptr -3F0h

  push    ebp
  mov     ebp, esp
  sub     esp, 3F8h
  call    _getchar
  sub     esp, 0Ch
  push    offset aEnterTheHardCo ; "Enter the hard-coded password (option 1"...
  call    _puts
  add     esp, 10h
  sub     esp, 8
  lea     eax, [ebp+s1]
  push    eax
  push    offset asc_804923E ; "%[^\n]"
  call    ___isoc99_scanf
  add     esp, 10h
  sub     esp, 8
  lea     eax, [ebp+s1]
  push    eax
  push    offset format    ; "Your input hard-coded password: %s\n"
  call    _printf
  add     esp, 10h
  sub     esp, 8
  push    offset s2        ; "Do what is right, not what is easy"
  lea     eax, [ebp+s1]
  push    eax              ; s1
  call    _strcmp
  add     esp, 10h
  test    eax, eax
  jnz     short loc_80486FE
  call    success_1
  jmp     short loc_8048703
-----------------------------------------------------------------

                           ; CODE XREF: hardCode+65↑j
  call    failed

                           ; CODE XREF: hardCode+6C↑j
  nop
  leave
000006B1 080486B1: hardCode+21
```

11:31 PM 4/28/2022

Get the input in ebp+s1 -> save the input in eax with lea.

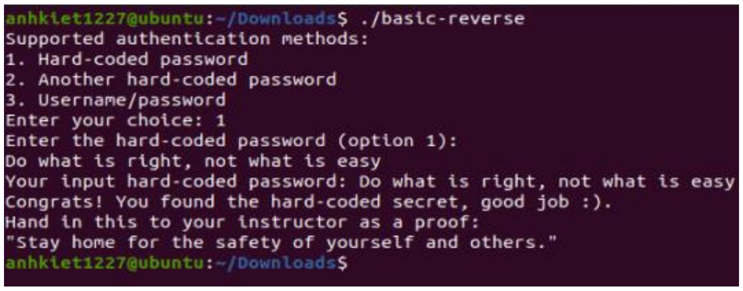
Push eax and push offset s2 used to push 2 strings into strcmp function to check the password. The function used the s2 to compare with the s1, so maybe the password is the s2. Let's check the s2!



```
.rodata:08049268 ; char s2[]
.rodata:08049268 s2              db 'Do what is right, not what is easy',0
.rodata:08049268                                 ; DATA XREF: hardCode+4F↑o
```

The s2 show: 'Do what is right, not what is easy'

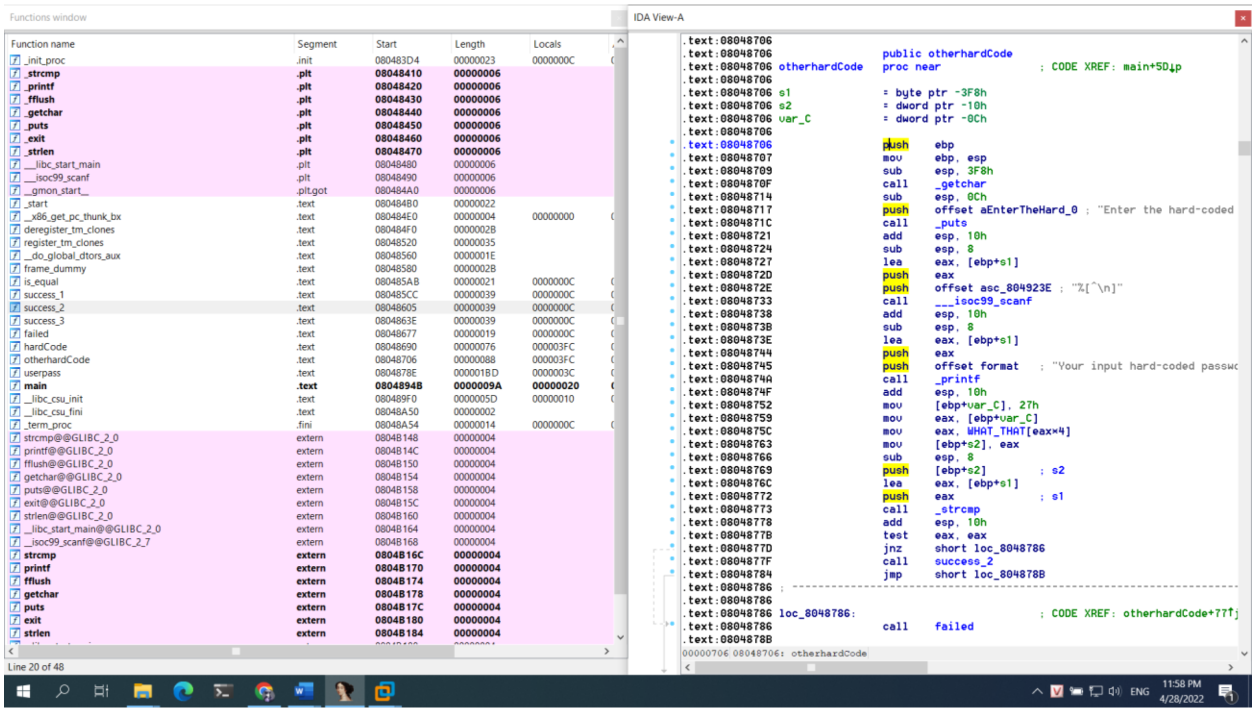
Maybe it is the password. Go to Linux and check it!



```
anhkiet1227@ubuntu:~/Downloads$ ./basic-reverse
Supported authentication methods:
1. Hard-coded password
2. Another hard-coded password
3. Username/password
Enter your choice: 1
Enter the hard-coded password (option 1):
Do what is right, not what is easy
Your input hard-coded password: Do what is right, not what is easy
Congrats! You found the hard-coded secret, good job :).
Hand in this to your instructor as a proof:
"Stay home for the safety of yourself and others."
anhkiet1227@ubuntu:~/Downloads$
```

That is the password!

Task 2: Other times other ways

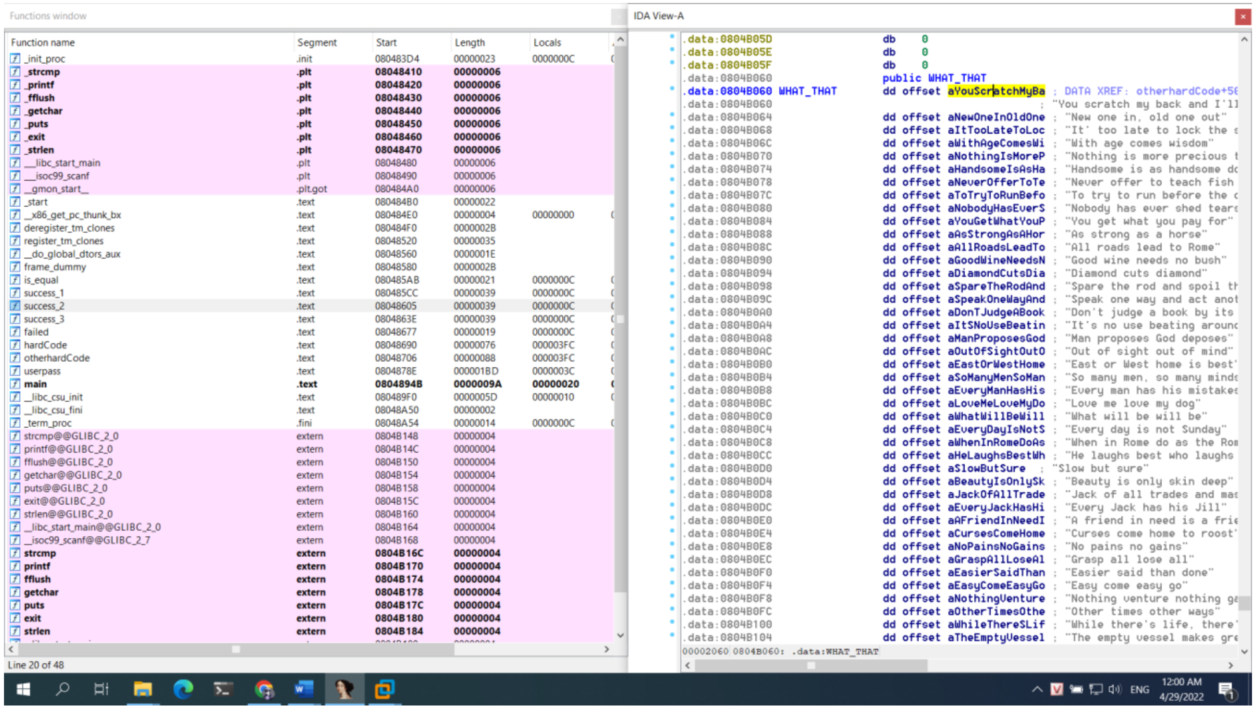


```
Functions window
Function name                  Segment   Start     Length    Locals
_init_proc                     .init     080483D4  00000023  0000000C
_strcmp                        .plt      08048410  00000006
_printf                        .plt      08048420  00000006
_fflush                        .plt      08048430  00000006
_getchar                       .plt      08048440  00000006
_puts                          .plt      08048450  00000006
_exit                          .plt      08048460  00000006
_strlen                        .plt      08048470  00000006
___libc_start_main             .plt      08048480  00000006
___isoc99_scanf                .plt      08048490  00000006
__gmon_start__                 .plt.got  080484A0  00000006
_start                         .text     080484B0  00000022
__x86_get_pc_thunk_bx          .text     080484E0  00000004  00000000
deregister_tm_clones           .text     080484F0  0000002B
register_tm_clones             .text     08048520  00000035
__do_global_dtors_aux          .text     08048560  0000001E
frame_dummy                    .text     08048580  0000002B
is_equal                       .text     080485AB  00000021  0000000C
success_1                      .text     080485CC  00000039  0000000C
success_2                      .text     08048605  00000039  0000000C
success_3                      .text     0804863E  00000039  0000000C
failed                         .text     08048677  00000019  0000000C
hardCode                       .text     08048690  00000076  000003FC
otherhardCode                  .text     08048706  00000088  000003FC
userpass                       .text     0804878E  000001BD  0000003C
main                           .text     0804894B  0000009A  00000020
__libc_csu_init                .text     080489F0  0000005D  00000010
__libc_csu_fini                .text     08048A50  00000002
_term_proc                     .fini     08048A54  00000014  0000000C
strcmp@@GLIBC_2_0              extern    0804B148  00000004
printf@@GLIBC_2_0              extern    0804B14C  00000004
fflush@@GLIBC_2_0              extern    0804B150  00000004
getchar@@GLIBC_2_0             extern    0804B154  00000004
puts@@GLIBC_2_0                extern    0804B158  00000004
exit@@GLIBC_2_0                extern    0804B15C  00000004
strlen@@GLIBC_2_0              extern    0804B160  00000004
__libc_start_main@@GLIBC_2_0   extern    0804B164  00000004
__isoc99_scanf@@GLIBC_2_7      extern    0804B168  00000004
strcmp                         extern    0804B16C  00000004
printf                         extern    0804B170  00000004
fflush                         extern    0804B174  00000004
getchar                        extern    0804B178  00000004
puts                           extern    0804B17C  00000004
exit                           extern    0804B180  00000004
strlen                         extern    0804B184  00000004
Line 20 of 48
```

```
IDA View-A
.text:08048706
.text:08048706                 public otherhardCode
.text:08048706 otherhardCode   proc near               ; CODE XREF: main+5D↓p
.text:08048706
.text:08048706 s1              = byte ptr -3F8h
.text:08048706 s2              = dword ptr -10h
.text:08048706 var_C           = dword ptr -0Ch
.text:08048706
.text:08048706                 push    ebp
.text:08048707                 mov     ebp, esp
.text:08048709                 sub     esp, 3F8h
.text:0804870F                 call    _getchar
.text:08048714                 sub     esp, 0Ch
.text:08048717                 push    offset aEnterTheHard_0 ; "Enter the hard-coded
.text:0804871C                 call    _puts
.text:08048721                 add     esp, 10h
.text:08048724                 sub     esp, 8
.text:08048727                 lea     eax, [ebp+s1]
.text:0804872D                 push    eax
.text:0804872E                 push    offset asc_804923E ; "%[^\n]"
.text:08048733                 call    ___isoc99_scanf
.text:08048738                 add     esp, 10h
.text:0804873B                 sub     esp, 8
.text:0804873E                 lea     eax, [ebp+s1]
.text:08048744                 push    eax
.text:08048745                 push    offset format   ; "Your input hard-coded passwc
.text:0804874A                 call    _printf
.text:0804874F                 add     esp, 10h
.text:08048752                 mov     [ebp+var_C], 27h
.text:08048759                 mov     eax, [ebp+var_C]
.text:0804875C                 mov     eax, WHAT_THAT[eax*4]
.text:08048763                 mov     [ebp+s2], eax
.text:08048766                 sub     esp, 8
.text:08048769                 push    [ebp+s2]        ; s2
.text:0804876C                 lea     eax, [ebp+s1]
.text:08048772                 push    eax             ; s1
.text:08048773                 call    _strcmp
.text:08048778                 add     esp, 10h
.text:0804877B                 test    eax, eax
.text:0804877D                 jnz     short loc_8048786
.text:0804877F                 call    success_2
.text:08048784                 jmp     short loc_804878B
.text:08048786 ; ---------------------------------------------------------------
.text:08048786
.text:08048786 loc_8048786:                            ; CODE XREF: otherhardCode+77↑j
.text:08048786                 call    failed
.text:0804878B
00000706 08048706: otherhardCode
```

11:58 PM 4/28/2022

Push the eax (get data from WHAT_THAT) and push [ebp + s2]. Maybe the function wants to get the input and check the input with something in WHAT_THAT



```
Functions window
Function name                  Segment   Start     Length    Locals
_init_proc                     .init     080483D4  00000023  0000000C
_strcmp                        .plt      08048410  00000006
_printf                        .plt      08048420  00000006
_fflush                        .plt      08048430  00000006
_getchar                       .plt      08048440  00000006
_puts                          .plt      08048450  00000006
_exit                          .plt      08048460  00000006
_strlen                        .plt      08048470  00000006
___libc_start_main             .plt      08048480  00000006
___isoc99_scanf                .plt      08048490  00000006
__gmon_start__                 .plt.got  080484A0  00000006
_start                         .text     080484B0  00000022
__x86_get_pc_thunk_bx          .text     080484E0  00000004  00000000
deregister_tm_clones           .text     080484F0  0000002B
register_tm_clones             .text     08048520  00000035
__do_global_dtors_aux          .text     08048560  0000001E
frame_dummy                    .text     08048580  0000002B
is_equal                       .text     080485AB  00000021  0000000C
success_1                      .text     080485CC  00000039  0000000C
success_2                      .text     08048605  00000039  0000000C
success_3                      .text     0804863E  00000039  0000000C
failed                         .text     08048677  00000019  0000000C
hardCode                       .text     08048690  00000076  000003FC
otherhardCode                  .text     08048706  00000088  000003FC
userpass                       .text     0804878E  000001BD  0000003C
main                           .text     0804894B  0000009A  00000020
__libc_csu_init                .text     080489F0  0000005D  00000010
__libc_csu_fini                .text     08048A50  00000002
_term_proc                     .fini     08048A54  00000014  0000000C
strcmp@@GLIBC_2_0              extern    0804B148  00000004
printf@@GLIBC_2_0              extern    0804B14C  00000004
fflush@@GLIBC_2_0              extern    0804B150  00000004
getchar@@GLIBC_2_0             extern    0804B154  00000004
puts@@GLIBC_2_0                extern    0804B158  00000004
exit@@GLIBC_2_0                extern    0804B15C  00000004
strlen@@GLIBC_2_0              extern    0804B160  00000004
__libc_start_main@@GLIBC_2_0   extern    0804B164  00000004
__isoc99_scanf@@GLIBC_2_7      extern    0804B168  00000004
strcmp                         extern    0804B16C  00000004
printf                         extern    0804B170  00000004
fflush                         extern    0804B174  00000004
getchar                        extern    0804B178  00000004
puts                           extern    0804B17C  00000004
exit                           extern    0804B180  00000004
strlen                         extern    0804B184  00000004
Line 20 of 48
```

```
IDA View-A
.data:0804B05D                 db    0
.data:0804B05E                 db    0
.data:0804B05F                 db    0
.data:0804B060                 public WHAT_THAT
.data:0804B060 WHAT_THAT       dd offset aYouScratchMyBa ; DATA XREF: otherhardCode+56
.data:0804B060                                         ; "You scratch my back and I'll
.data:0804B064                 dd offset aNewOneInOldOne ; "New one in, old one out"
.data:0804B068                 dd offset aItTooLateToLoc ; "It' too late to lock the s
.data:0804B06C                 dd offset aWithAgeComesWi ; "With age comes wisdom"
.data:0804B070                 dd offset aNothingIsMoreP ; "Nothing is more precious t
.data:0804B074                 dd offset aHandsomeIsAsHa ; "Handsome is as handsome dc
.data:0804B078                 dd offset aNeverOfferToTe ; "Never offer to teach fish
.data:0804B07C                 dd offset aToTryToRunBefo ; "To try to run before the c
.data:0804B080                 dd offset aNobodyHasEverS ; "Nobody has ever shed tears
.data:0804B084                 dd offset aYouGetWhatYouP ; "You get what you pay for"
.data:0804B088                 dd offset aAsStrongAsAHor ; "As strong as a horse"
.data:0804B08C                 dd offset aAllRoadsLeadTo ; "All roads lead to Rome"
.data:0804B090                 dd offset aGoodWineNeedsN ; "Good wine needs no bush"
.data:0804B094                 dd offset aDiamondCutsDia ; "Diamond cuts diamond"
.data:0804B098                 dd offset aSpareTheRodAnd ; "Spare the rod and spoil th
.data:0804B09C                 dd offset aSpeakOneWayAnd ; "Speak one way and act anot
.data:0804B0A0                 dd offset aDonTJudgeABook ; "Don't judge a book by its
.data:0804B0A4                 dd offset aItSNoUseBeatin ; "It's no use beating arounc
.data:0804B0A8                 dd offset aManProposesGod ; "Man proposes God deposes"
.data:0804B0AC                 dd offset aOutOfSightOutO ; "Out of sight out of mind"
.data:0804B0B0                 dd offset aEastOrWestHome ; "East or West home is best"
.data:0804B0B4                 dd offset aSoManyMenSoMan ; "So many men, so many minds
.data:0804B0B8                 dd offset aEveryManHasHis ; "Every man has his mistakes
.data:0804B0BC                 dd offset aLoveMeLoveMyDo ; "Love me love my dog"
.data:0804B0C0                 dd offset aWhatWillBeWill ; "What will be will be"
.data:0804B0C4                 dd offset aEveryDayIsNotS ; "Every day is not Sunday"
.data:0804B0C8                 dd offset aWhenInRomeDoAs ; "When in Rome do as the Rom
.data:0804B0CC                 dd offset aHeLaughsBestWh ; "He laughs best who laughs
.data:0804B0D0                 dd offset aSlowButSure  ; "Slow but sure"
.data:0804B0D4                 dd offset aBeautyIsOnlySk ; "Beauty is only skin deep"
.data:0804B0D8                 dd offset aJackOfAllTrade ; "Jack of all trades and mas
.data:0804B0DC                 dd offset aEveryJackHasHi ; "Every Jack has his Jill"
.data:0804B0E0                 dd offset aAFriendInNeedI ; "A friend in need is a frie
.data:0804B0E4                 dd offset aCursesComeHome ; "Curses come home to roost"
.data:0804B0E8                 dd offset aNoPainsNoGains ; "No pains no gains"
.data:0804B0EC                 dd offset aGraspAllLoseAl ; "Grasp all lose all"
.data:0804B0F0                 dd offset aEasierSaidThan ; "Easier said than done"
.data:0804B0F4                 dd offset aEasyComeEasyGo ; "Easy come easy go"
.data:0804B0F8                 dd offset aNothingVenture ; "Nothing venture nothing ga
.data:0804B0FC                 dd offset aOtherTimesOthe ; "Other times other ways"
.data:0804B100                 dd offset aWhileThereSLif ; "While there's life, there'
.data:0804B104                 dd offset aTheEmptyVessel ; "The empty vessel makes gre
00002060 0804B060: .data:WHAT_THAT
```

12:00 AM 4/29/2022

There are many things in WHAT_THAT. Check again in otherhardCore:

Transfer 27(hex) into [ebp + var_C]

Then save 27(hex) into eax with [ebp + var_C]

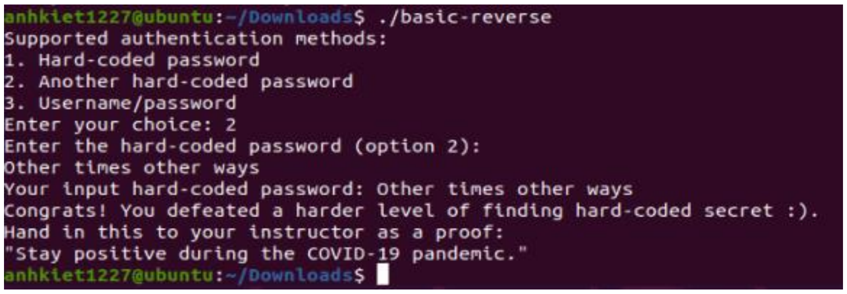
Then get the data in the place eax*4 in WHAT_THAT that means 27*4 = 9C

Get into WHAT_THAT, the first address is 0804B060 then add with 9C = 0804B0FC

Maybe the password in the place with this address is 0804B0FC

‘Other times other ways’

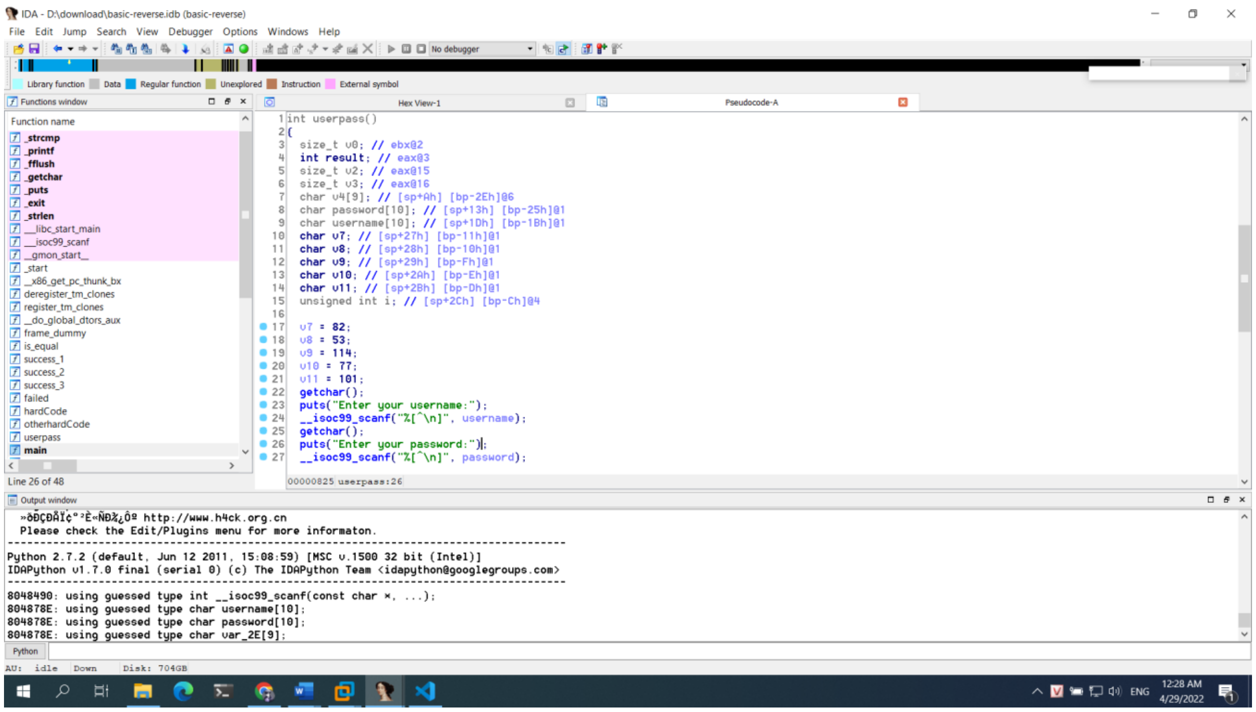
Check it in Linux



```
anhkiet1227@ubuntu:~/Downloads$ ./basic-reverse
Supported authentication methods:
1. Hard-coded password
2. Another hard-coded password
3. Username/password
Enter your choice: 2
Enter the hard-coded password (option 2):
Other times other ways
Your input hard-coded password: Other times other ways
Congrats! You defeated a harder level of finding hard-coded secret :).
Hand in this to your instructor as a proof:
"Stay positive during the COVID-19 pandemic."
anhkiet1227@ubuntu:~/Downloads$
```

That is the password

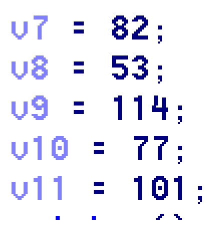
Task 3: 0504?2T>L



```
int userpass()
{
  size_t v0; // ebx@2
  int result; // eax@3
  size_t v2; // eax@15
  size_t v3; // eax@16
  char v4[9]; // [sp+Ah] [bp-2Eh]@6
  char password[10]; // [sp+13h] [bp-25h]@1
  char username[10]; // [sp+1Dh] [bp-1Bh]@1
  char v7; // [sp+27h] [bp-11h]@1
  char v8; // [sp+28h] [bp-10h]@1
  char v9; // [sp+29h] [bp-Fh]@1
  char v10; // [sp+2Ah] [bp-Eh]@1
  char v11; // [sp+2Bh] [bp-Dh]@1
  unsigned int i; // [sp+2Ch] [bp-Ch]@4

  v7 = 82;
  v8 = 53;
  v9 = 114;
  v10 = 77;
  v11 = 101;
  getchar();
  puts("Enter your username:");
  __isoc99_scanf("%[^\n]", username);
  getchar();
  puts("Enter your password:");
  __isoc99_scanf("%[^\n]", password);
```

```
 »ðÐÇÐÅÏ¢º²È«ÑÐ¾¿Ôº http://www.h4ck.org.cn
 Please check the Edit/Plugins menu for more informaton.
---------------------------------------------------------------------------------------
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)]
IDAPython v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
---------------------------------------------------------------------------------------
8048490: using guessed type int __isoc99_scanf(const char *, ...);
804878E: using guessed type char username[10];
804878E: using guessed type char password[10];
804878E: using guessed type char var_2E[9];
```

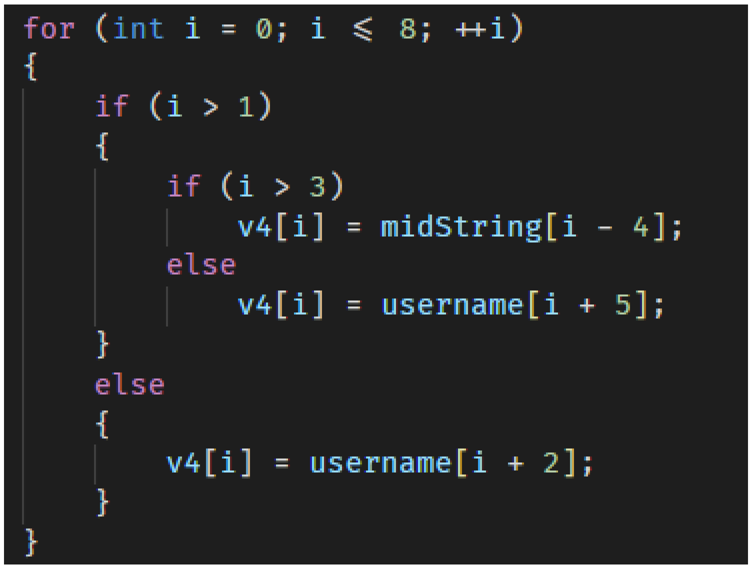
First thing we need to get the pseudocode



```
v7 = 82;
v8 = 53;
v9 = 114;
v10 = 77;
v11 = 101;
```

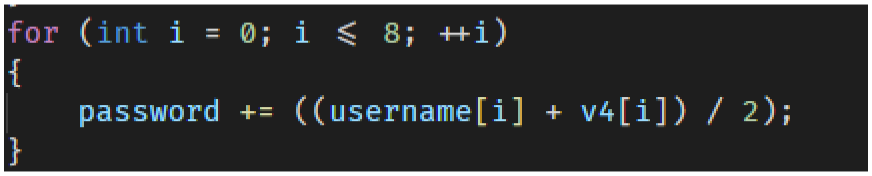
The midString from v7 to v11 is the ascii number so we decide to change to the text is ‘R5rMe’

The username get from our ID is ‘0605-0704’



```
for (int i = 0; i ⩽ 8; ++i)
{
    if (i > 1)
    {
        if (i > 3)
            v4[i] = midString[i - 4];
        else
            v4[i] = username[i + 5];
    }
    else
    {
        v4[i] = username[i + 2];
    }
}
```

With the v4 we get it from the pseudocode and have a little change with v7 to v11 become the midString with ascii decoded is ‘R5rMe’

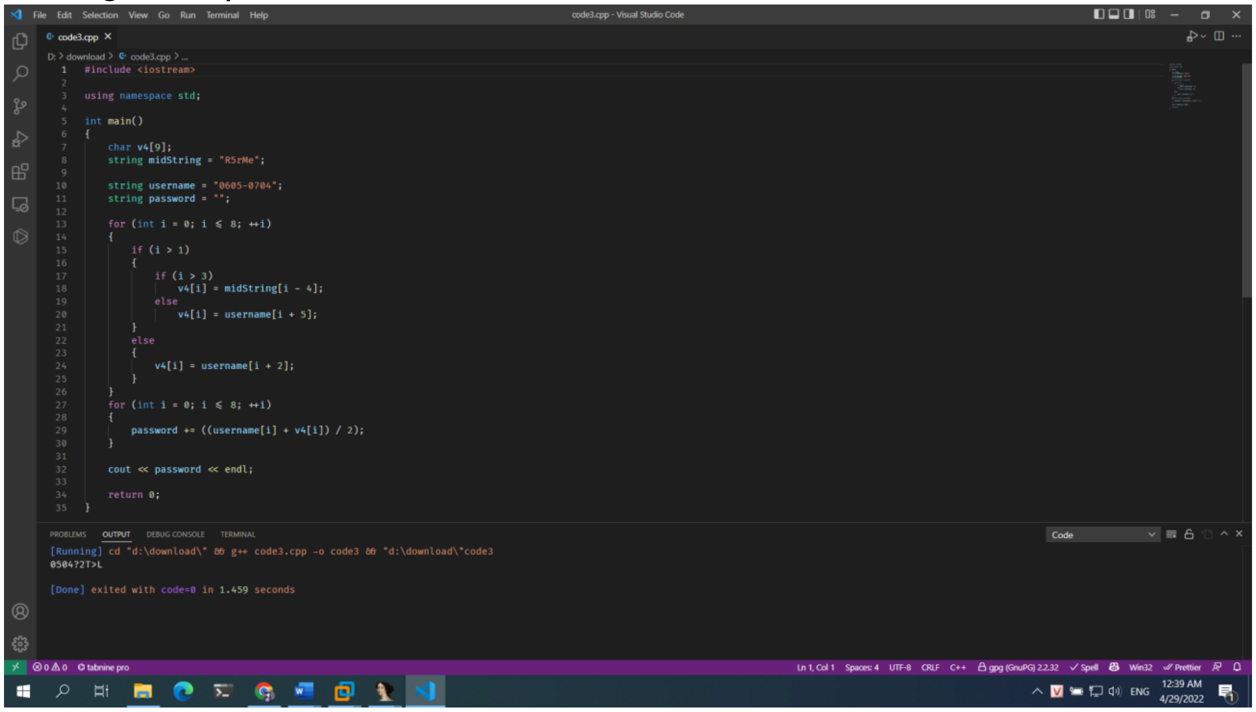


```
for (int i = 0; i ⩽ 8; ++i)
{
    password += ((username[i] + v4[i]) / 2);
}
```

And the password gets from the (username[i] + v4[i]) / 2 transfering from

(username[i] + v4[i]) / 2 != password[i]

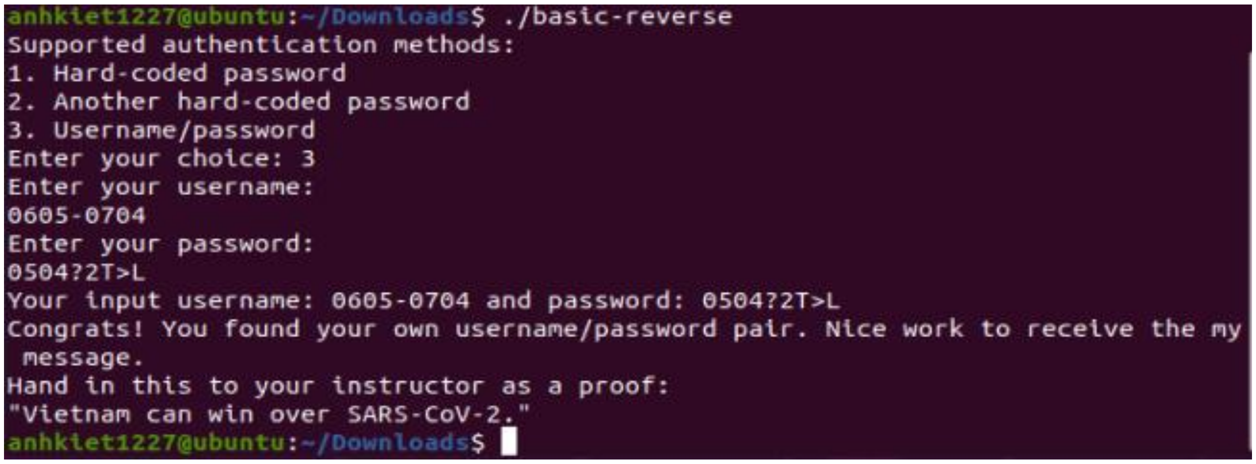
Then get the password with code: ‘0504?2T>L’



```cpp
#include <iostream>

using namespace std;

int main()
{
    char v4[9];
    string midString = "R5rMe";

    string username = "0605-0704";
    string password = "";

    for (int i = 0; i <= 8; ++i)
    {
        if (i > 1)
        {
            if (i > 3)
                v4[i] = midString[i - 4];
            else
                v4[i] = username[i + 5];
        }
        else
        {
            v4[i] = username[i + 2];
        }
    }
    for (int i = 0; i <= 8; ++i)
    {
        password += ((username[i] + v4[i]) / 2);
    }

    cout << password << endl;

    return 0;
}
```

```
[Running] cd "d:\download\" && g++ code3.cpp -o code3 && "d:\download\"code3
0504?2T>L

[Done] exited with code=0 in 1.459 seconds
```

Check it in Linux



```
anhkiet1227@ubuntu:~/Downloads$ ./basic-reverse
Supported authentication methods:
1. Hard-coded password
2. Another hard-coded password
3. Username/password
Enter your choice: 3
Enter your username:
0605-0704
Enter your password:
0504?2T>L
Your input username: 0605-0704 and password: 0504?2T>L
Congrats! You found your own username/password pair. Nice work to receive the my
 message.
Hand in this to your instructor as a proof:
"Vietnam can win over SARS-CoV-2."
anhkiet1227@ubuntu:~/Downloads$ 
```

That is the password!