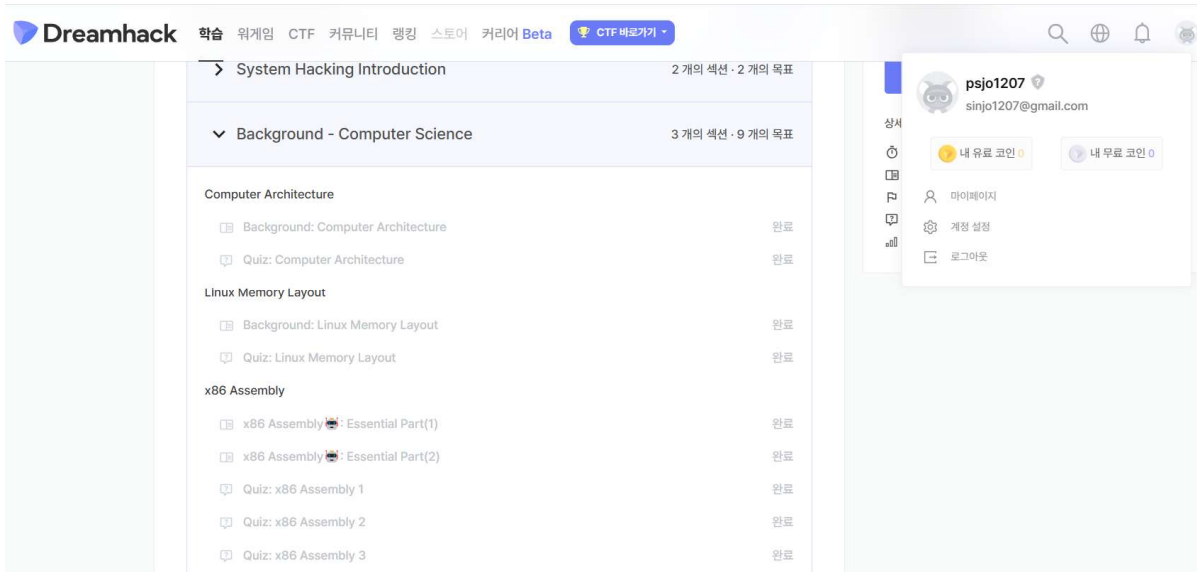


0408 대체과제

20223100 박신조

1번



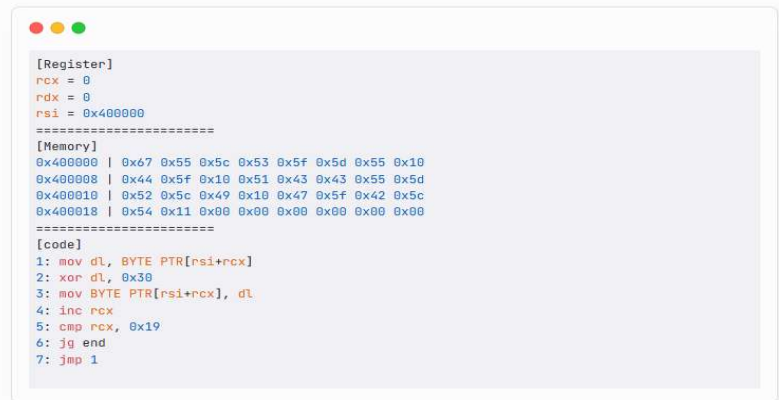
2번

```
어셈블리 2.py X
C: > Users > Lenovo > Desktop > 드림핵 > 어셈블리 2.py > ...
1  memory = [
2      0x67, 0x55, 0x5c, 0x53, 0x5f, 0x5d, 0x55, 0x10,
3      0x44, 0x5f, 0x10, 0x51, 0x43, 0x43, 0x55, 0x5d,
4      0x52, 0x5c, 0x49, 0x10, 0x47, 0x5f, 0x42, 0x5c,
5      0x54, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
6  ]
7
8  rcx = 0
9  while rcx <= 0x19:
10     memory[rcx] ^= 0x30
11     rcx += 1
12
13     print("\n== ASCII 문자열 ==")
14     rcx = 0
15     while rcx <= 0x19:
16         print(chr(memory[rcx]),end="")
17         rcx += 1

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Lenovo\Desktop\드림핵> ^C
PS C:\Users\Lenovo\Desktop\드림핵>
PS C:\Users\Lenovo\Desktop\드림핵> c++; cd 'c:\Users\Lenovo\Desktop\드림핵'; &
xtensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher'

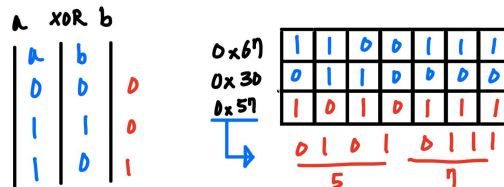
== ASCII 문자열 ==
Welcome to assembly world!
```

Q1. end로 점프하면 프로그램을 중단하자. 프로그램이 종료될 때, 0x400000부터 0x400019까지의 데이터를 조작하는 아스키 문자로 변환하면 어느 정도 문자열이 적용됩니까?



mov dl, BYTE PTR[rsi+rcx]
xor dl, 0x30

메모리[rsi + rcx]의 값 dl에 저장
dl = dl XOR 0x30
ex) 0x67 XOR 0x30 = 0x57



mov BYTE PTR[rsi+rcx], dl
inc rcx
cmp rcx, 0x19
jg end
jmp 1

다시 메모리에 저장
rcx += 1
총 25바이트인지 확인
25 넘으면 종료
아니면 루프 반복

```

1
2 import sys
3
4 memory = bytearray.fromhex(
5     '7233647930203730'
6     '2064336275363f00'
7 )
8
9 def write_n(ptr: bytes, length: int):
10     sys.stdout.buffer.write(ptr[:length])
11     sys.stdout.buffer.flush()
12
13 length = 0xf
14 ptr = memory
15 write_n(ptr, length)

```

16진수 문자열을 바이트 배열로 변환

ptr	출력할 데이터의 시작 주소
length	출력할 바이트 수
ptr[:length] =	ptr부터 length 만큼 출력

length = 0xf	출력 길이 15바이트
ptr = memory	0x400500에 해당하는 주소
write_n(ptr, length)	syscall과 동일

```
[Code]
main:
    push rbp
    mov rbp, rsp
    mov esi, 0xf
    mov rdi, 0x400500
    call 0x400497 <write_n>
    mov eax, 0x0
    pop rbp
    ret

write_n:
    push rbp
    mov rbp, rsp
    mov QWORD PTR [rbp-0x8], rdi
    mov DWORD PTR [rbp-0xc], esi
    xor rdx, rdx
    mov edx, DWORD PTR [rbp-0xc]
    mov rsi, QWORD PTR [rbp-0x8]
    mov rdi, 0x1
    mov rax, 0x1
    syscall
    pop rbp
    ret

=====

[Memory]
0x400500 | 0x3037207964343372
0x400508 | 0x003f367562336420
```

mov edx, DWORD PTR [rbp-0xc]	edx = 길이 (15)
mov rsi, QWORD PTR [rbp-0x8]	rsi = 버퍼 주소 (0x400500)
mov rdi, 0x1	rdi = 표준출력 (stdout)
mov rax, 0x1	rax = write 시스템 호출 번호
syscall	write(1, 0x400500, 15)