어 셈 블 리 어 2

Assembly Language 2

이다영

1. 레지스터

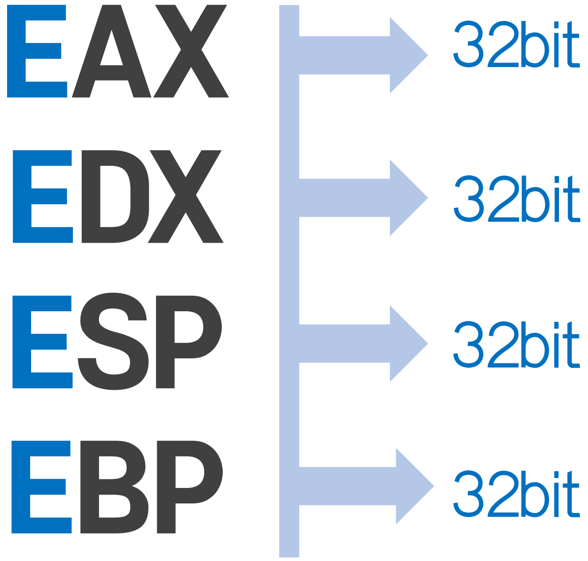
EAX

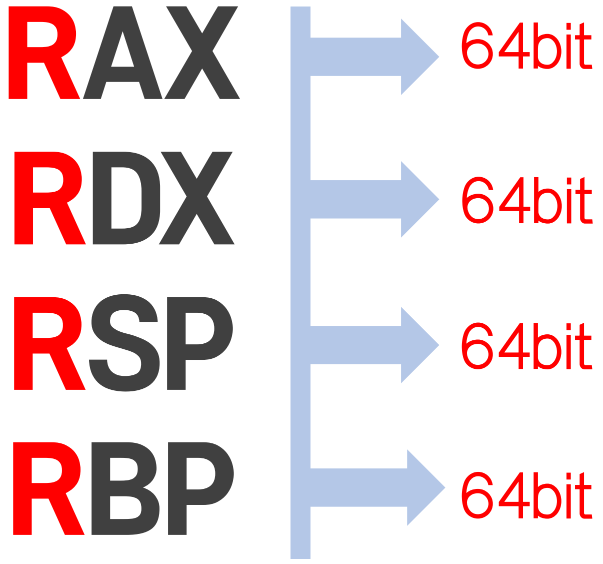
EDX

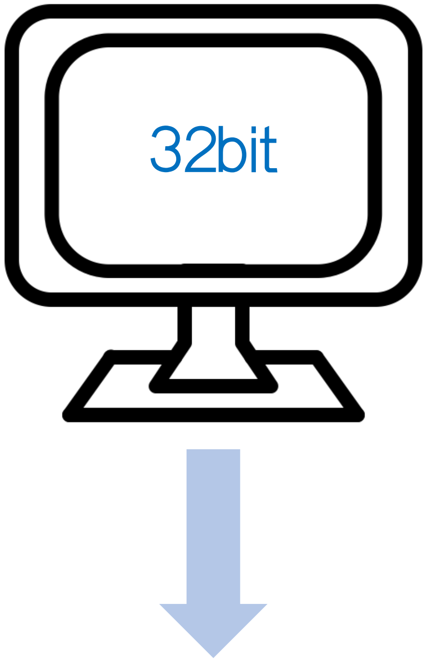
ESP

EBP

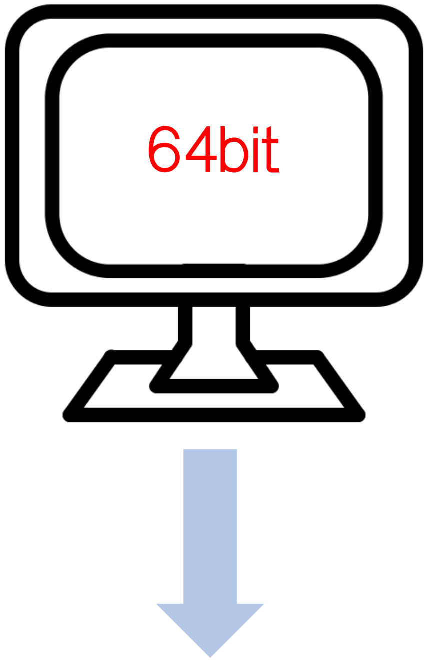
AX	16bit
DX	16bit
SP	16bit
BP	16bit


EAX
32bit
EDX
32bit
ESP
32bit
EBP
32bit


RAX
64bit
RDX
64bit
RSP
64bit
RBP
64bit

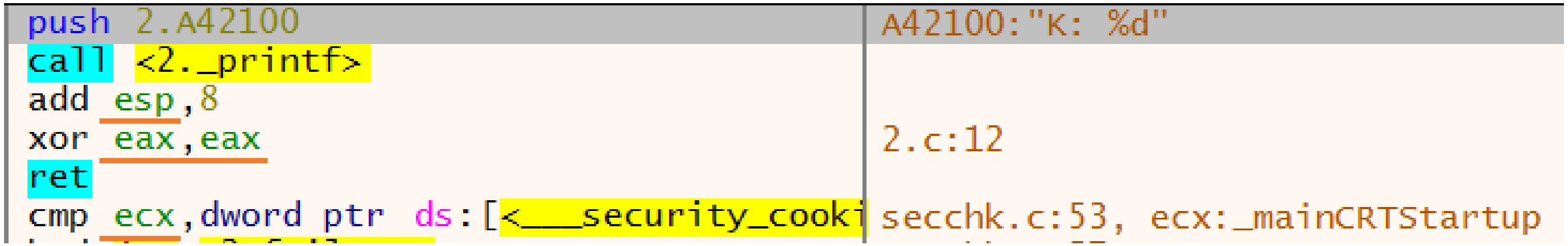


32비트 이하 프로그램



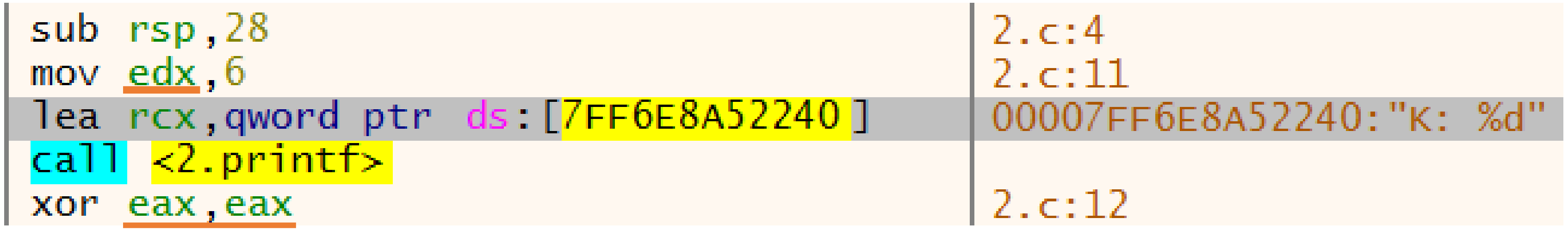
64비트 프로그램
32비트 프로그램

64bit 운영체제



```
push 2.A42100                                  A42100:"K: %d"
call <2._printf>
add  esp,8
xor  eax,eax                                   2.c:12
ret
cmp  ecx,dword ptr ds:[<___security_cooki      secchk.c:53, ecx:_mainCRTStartup
```

32비트 프로그램



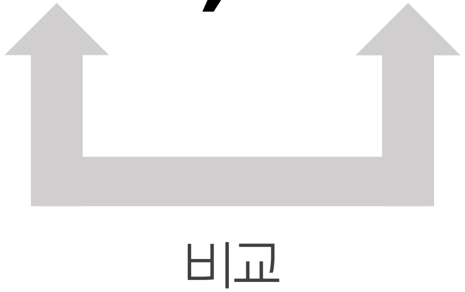
```
sub  rsp,28                                    2.c:4
mov  edx,6                                     2.c:11
lea  rcx,qword ptr ds:[7FF6E8A52240]           00007FF6E8A52240:"K: %d"
call <2.printf>
xor  eax,eax                                   2.c:12
```

64비트 프로그램

2.명령어

– 분기문

```
CMP EAX, 200
```



```
JMP [402000]
```

J -

조건 분기

JA	Jump Above	**JE**	Jump Equal
JB	Jump Below	**JNE**	Jump Not Equal

J -

조건 분기

JA	CMP $\overset{\text{Destination}}{\underline{a}} > \overset{\text{Source}}{\underline{b}}$	JE	CMP a == b
JB	CMP a < b	JNE	CMP a != b

3 If문 분석

함수 프롤로그&에필로그

프롤로그
: 함수 실행 준비과정

① PUSH EBP
② MOV EBP, ESP

스택에 RET(Return Address＝복귀 주소) 생성

① EBP를 스택에 넣음 → SFP(Stack Frame Pointer)를 생성해 호출한 함수의 EBP를 SFP에 백업

② ESP를 EBP에 복사

에필로그
: 함수 내에서의 수행을 마치고 처음 호출한 지점으로 돌아가기 위해 스택을 복원하는 과정

LEAVE

① MOV ESP, EBP

② POP EBP

① EBP를 ESP에 복사해 ESP를 EBP의 위치로 내림

② 현재 ESP가 가리키고 있는 SFP를 EBP에 넣어줌으로써 이전 함수로 돌아감

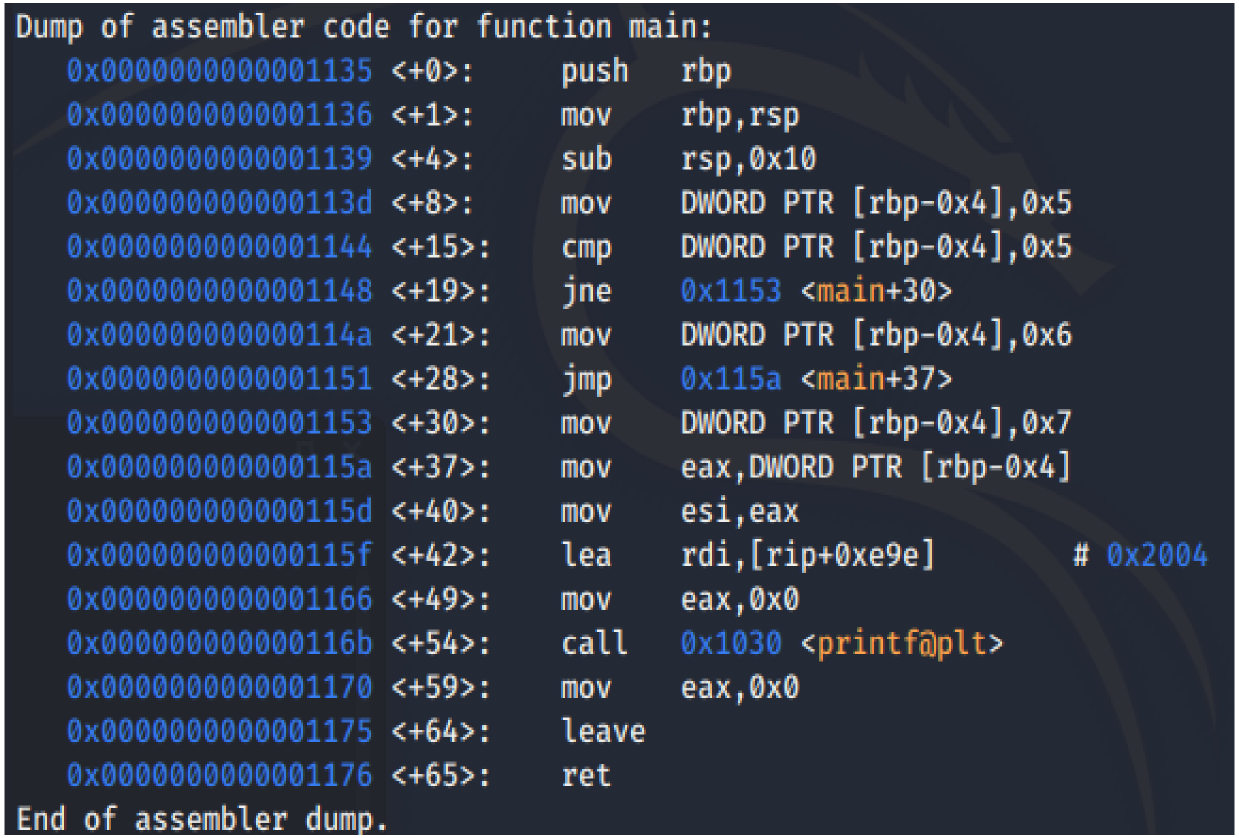
RET

① POP EIP
② JMP EIP

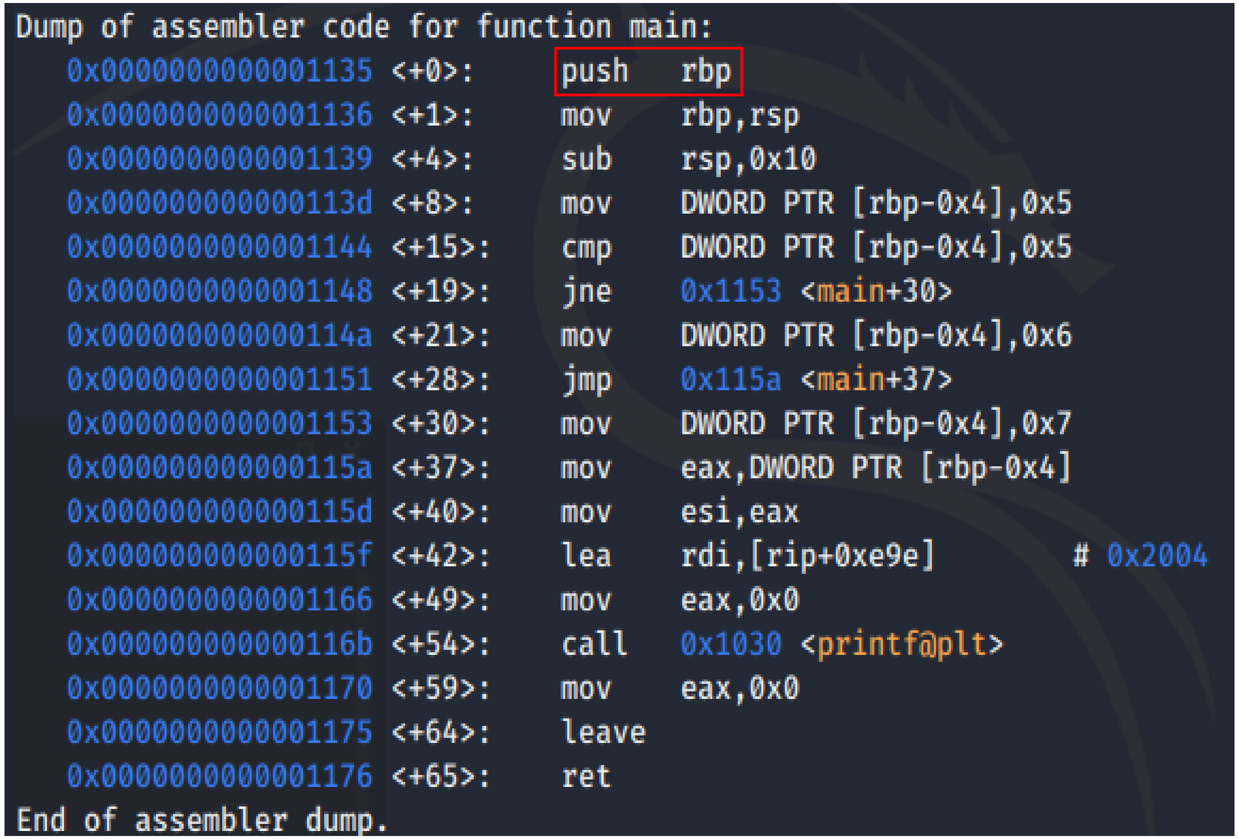
① 현재 ESP가 가리키고 있는 RET을 EIP에 넣음

② EIP로 이동 → 함수 호출 이후 명령을 계속 수행할 수 있도록 해줌

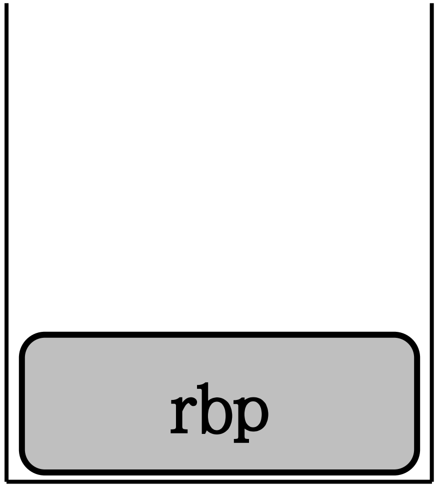
```c
int main(void)
{
int k = 5;
if (k == 5)
        k = 6;
else
        k = 7;
printf("K: %d", k);
return 0;
}
```

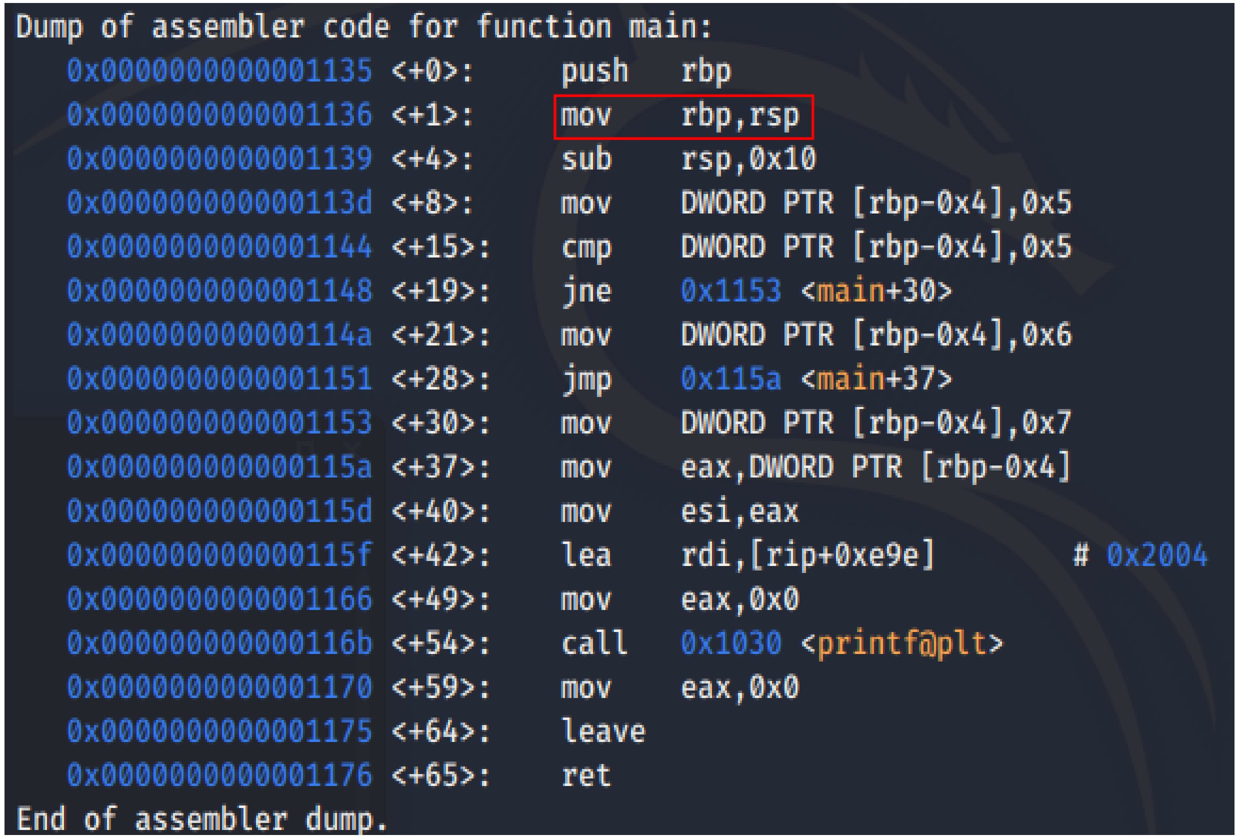
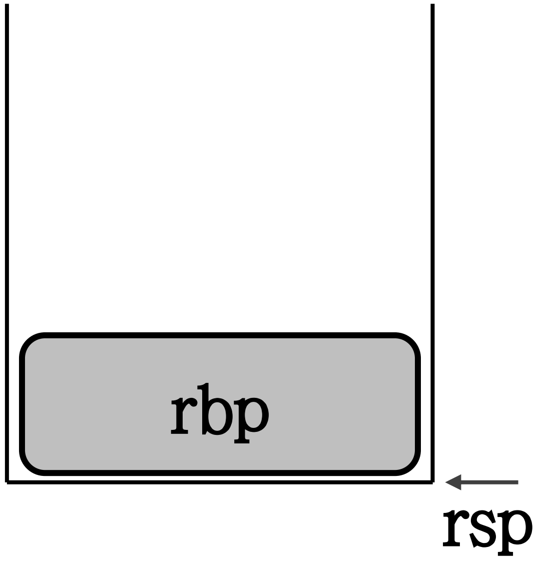


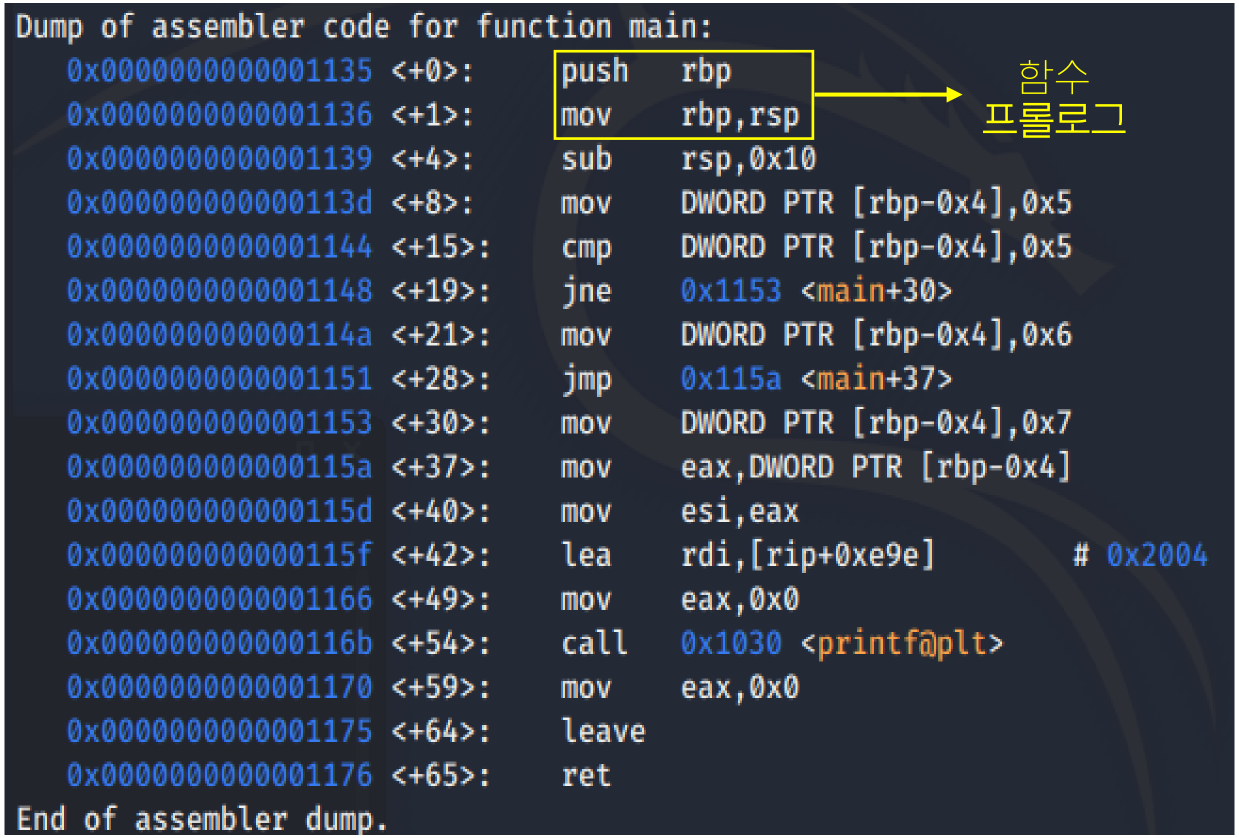
```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```



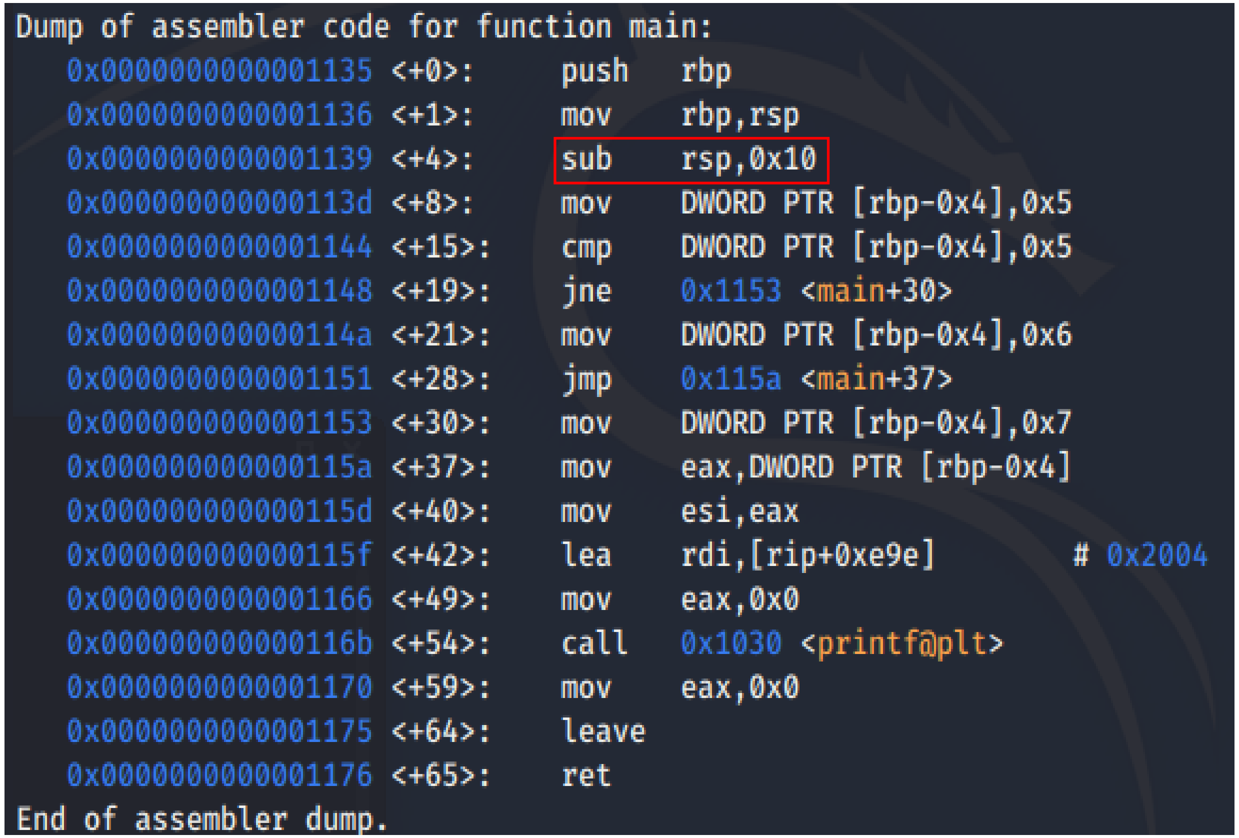
```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```



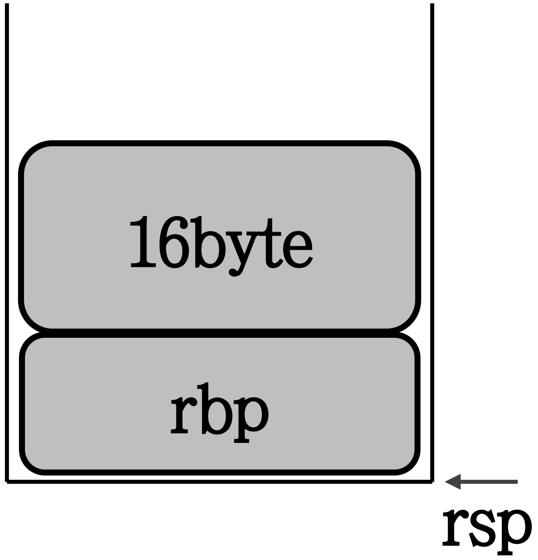

```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```


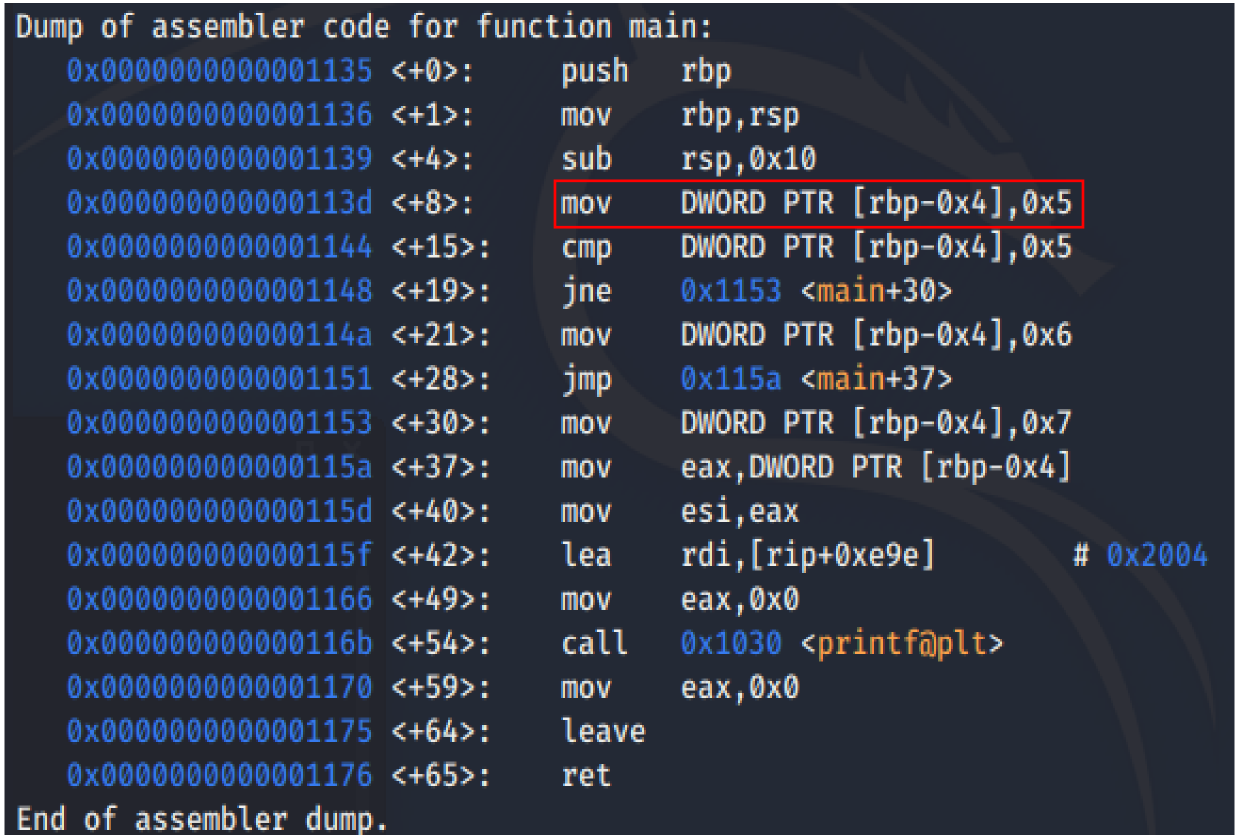


```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```



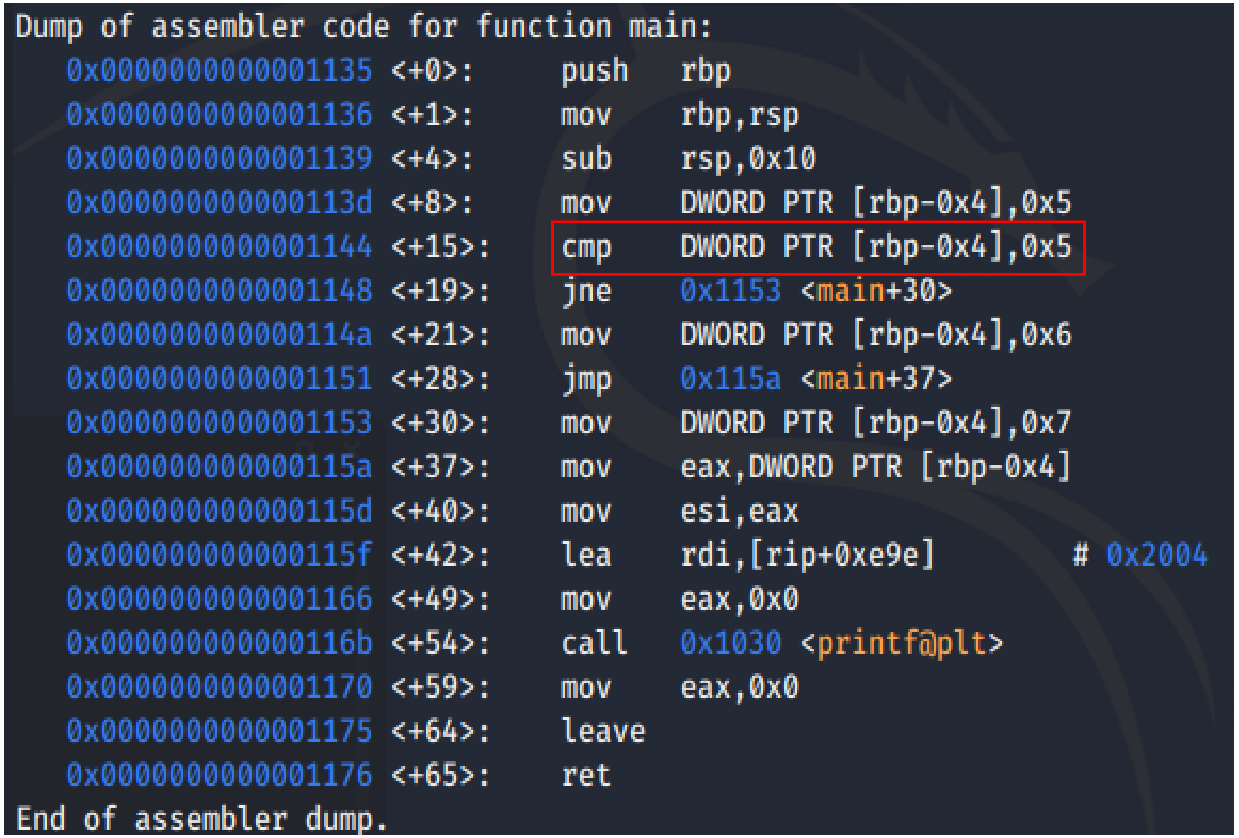
```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```





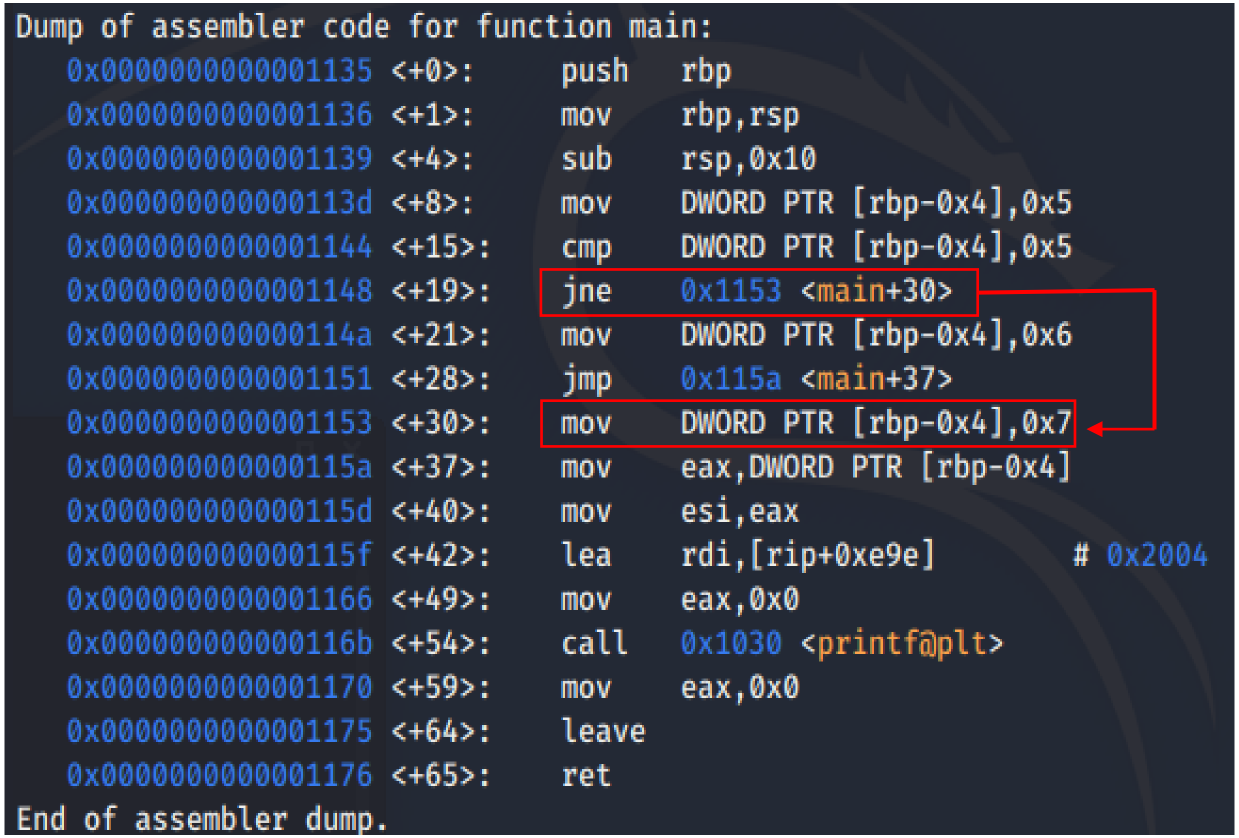
```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

```
int k = 5;
```



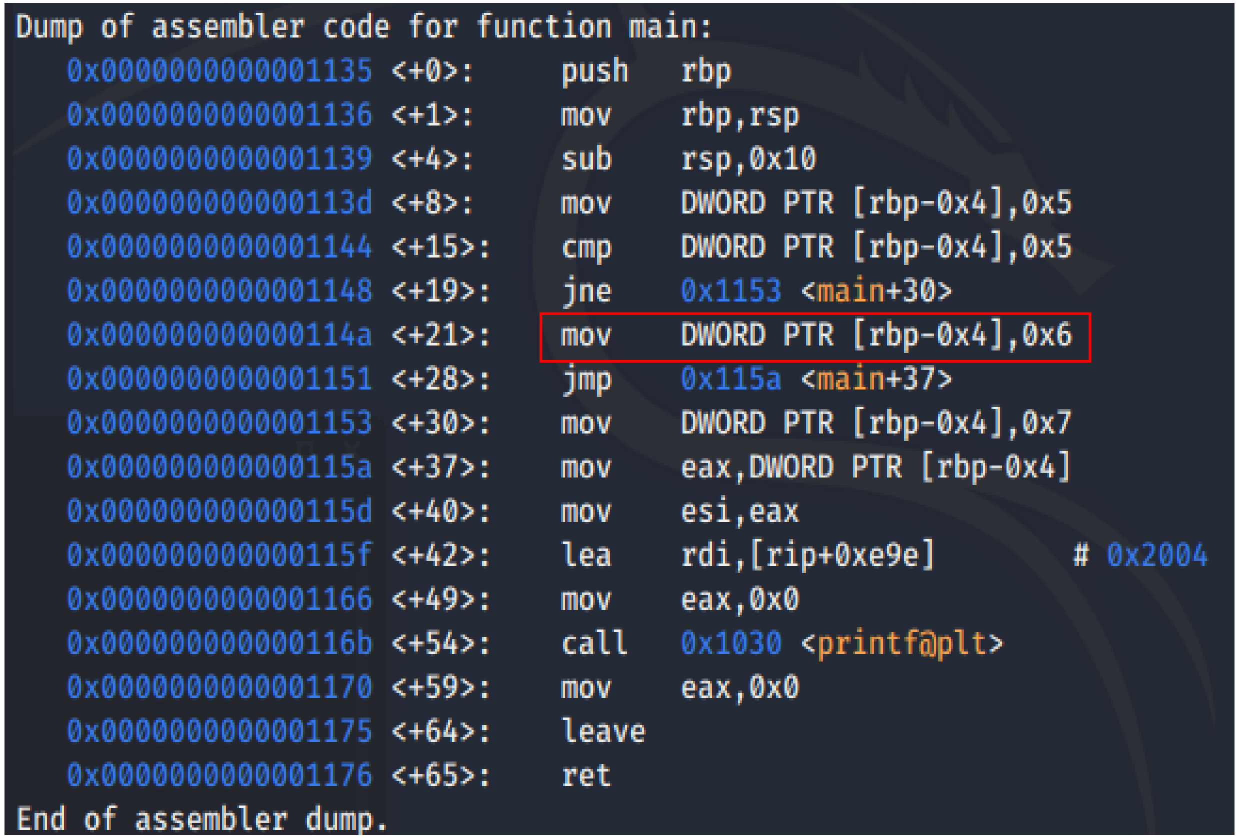
```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

if (k == 5)



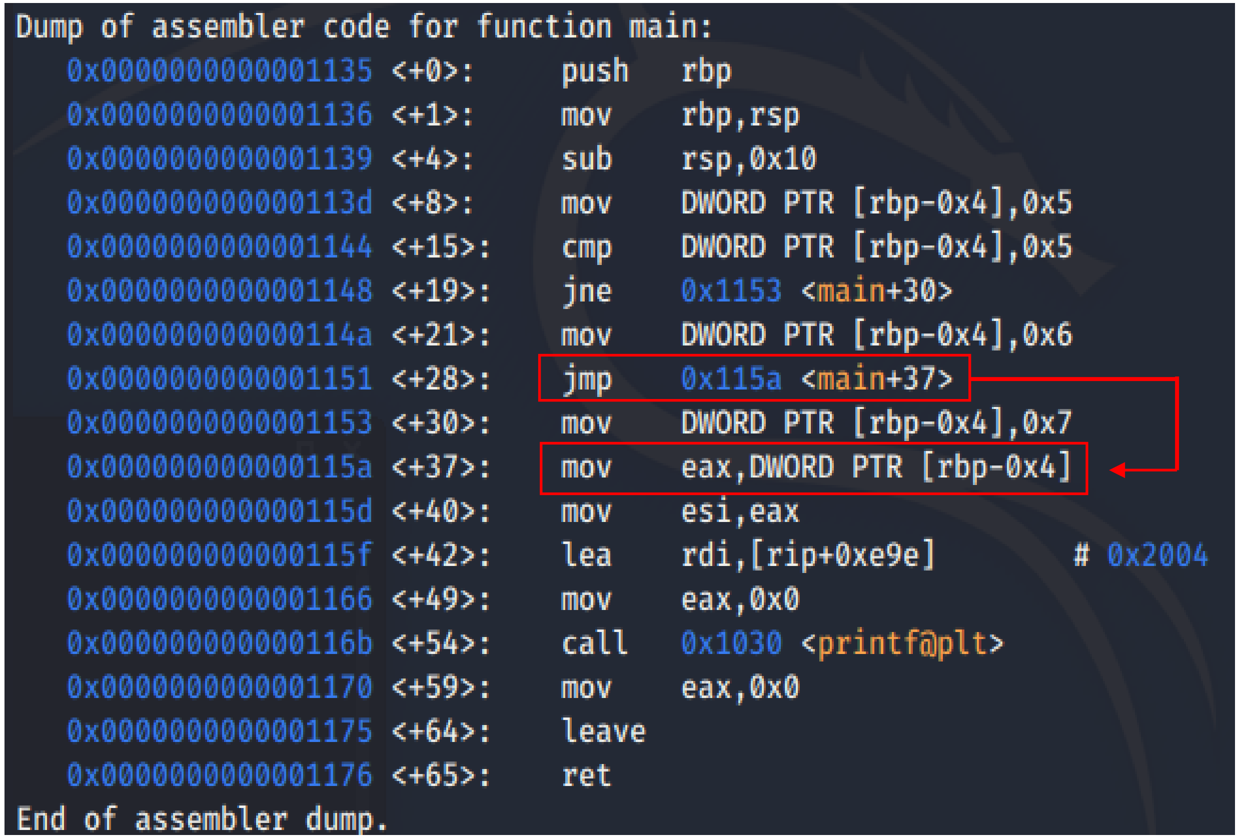
```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

```
else
k = 7;
```

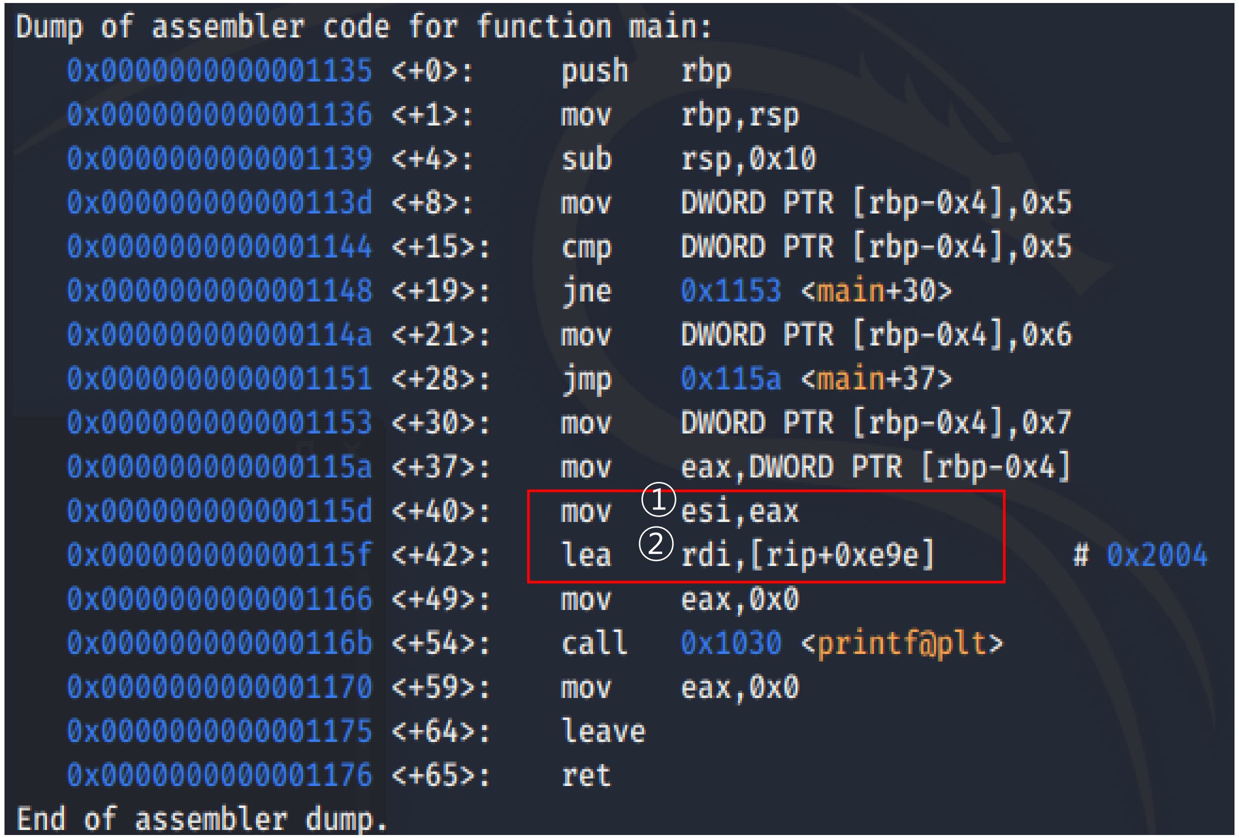


```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

```
k = 6;
```

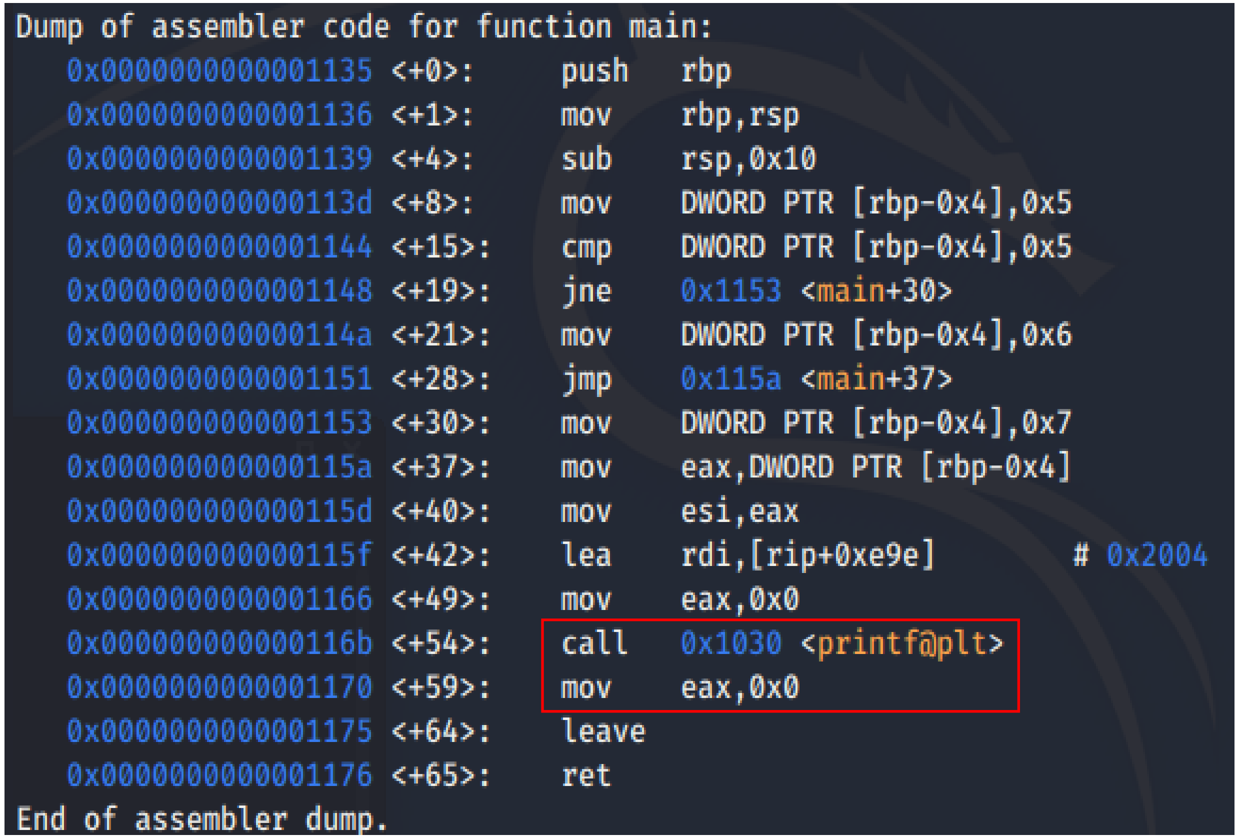


```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

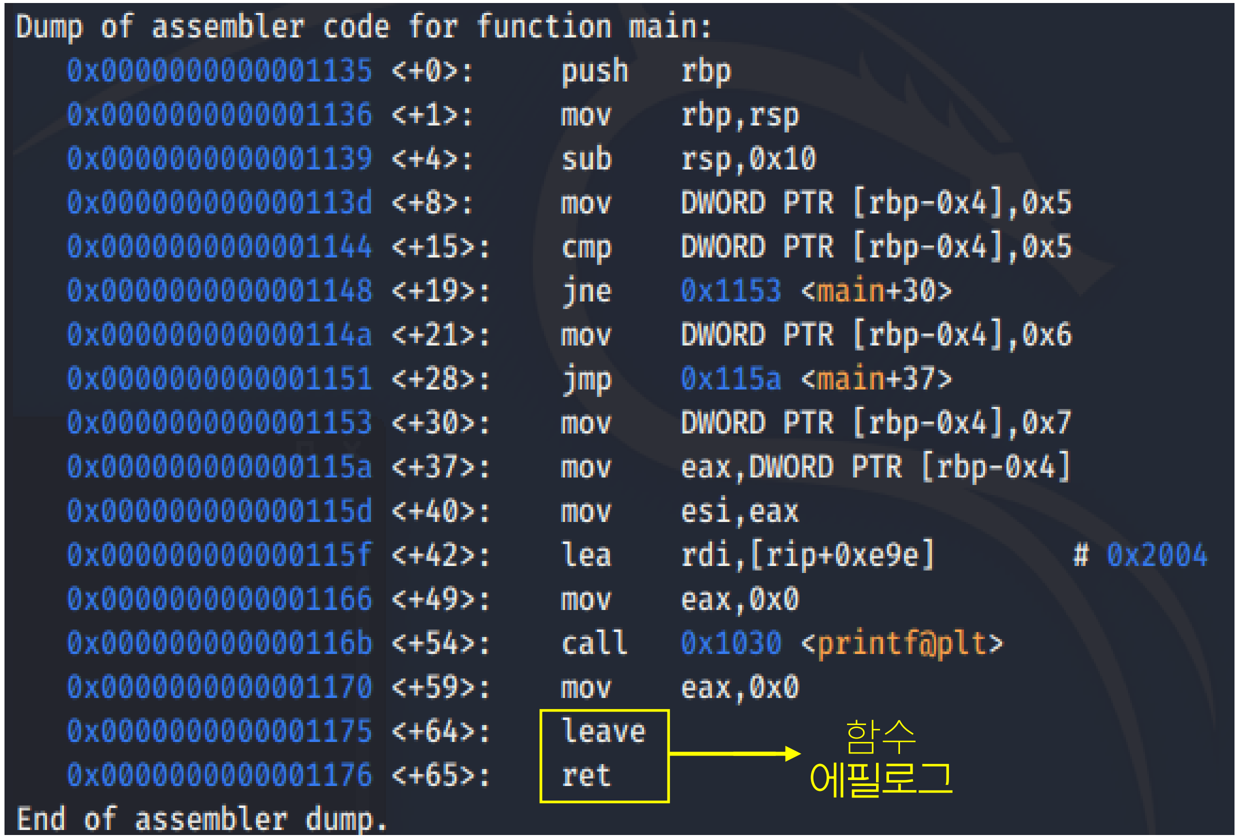


```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov  ① esi,eax
   0x000000000000115f <+42>:    lea  ② rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

printf($\overset{②}{\text{"K: \%d"}}$, $\overset{①}{\text{k}}$);



```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```



```
Dump of assembler code for function main:
   0x0000000000001135 <+0>:     push   rbp
   0x0000000000001136 <+1>:     mov    rbp,rsp
   0x0000000000001139 <+4>:     sub    rsp,0x10
   0x000000000000113d <+8>:     mov    DWORD PTR [rbp-0x4],0x5
   0x0000000000001144 <+15>:    cmp    DWORD PTR [rbp-0x4],0x5
   0x0000000000001148 <+19>:    jne    0x1153 <main+30>
   0x000000000000114a <+21>:    mov    DWORD PTR [rbp-0x4],0x6
   0x0000000000001151 <+28>:    jmp    0x115a <main+37>
   0x0000000000001153 <+30>:    mov    DWORD PTR [rbp-0x4],0x7
   0x000000000000115a <+37>:    mov    eax,DWORD PTR [rbp-0x4]
   0x000000000000115d <+40>:    mov    esi,eax
   0x000000000000115f <+42>:    lea    rdi,[rip+0xe9e]        # 0x2004
   0x0000000000001166 <+49>:    mov    eax,0x0
   0x000000000000116b <+54>:    call   0x1030 <printf@plt>
   0x0000000000001170 <+59>:    mov    eax,0x0
   0x0000000000001175 <+64>:    leave
   0x0000000000001176 <+65>:    ret
End of assembler dump.
```

THANK YOU.

감사합니다 .