Executable Before/After Relocation

```
0000000 <main>:
   . . .                    -4(수행시의 PC 보정값)
   e:   83 ec 04            sub    $0x4,%esp
  11:   e8 fc ff ff ff      call   12 <main+0x12>
                12: R_386_PC32 swap
  16:   83 c4 04            add    $0x4,%esp
   . . .
```

```
08048280 <main>:
  .   .   .
804828e:        83 ec 04                sub    $0x4,%esp
8048291:        e8 ?? ?? ?? ??          call   80502ca <swap>
8048296:        83 c4 04                add    $0x4,%esp
  .   .   .
```

(1) Call swap()을 수행할때의 PC의 값은 ?

- 다음 instruction은 add : 08048296 , 현재위치 (main+12) – (0x04) = (main+16)

(2) swap() 의 주소가 080502ca 라고 하면 relocation이 끝나서 ?? ?? ?? ?? 에 들어갈 값을 계산하시오.

- 080502ca – 08048296 =8034 => 34 80 00 00

Before Relocation (.text) swap.o

```
0000000000000000 <swap>:
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   48 c7 05 00 00 00 00    movq   $0x0,0x0(%rip)        # f <swap+0xf>
                        7: R_386_PC32        bufp1  -0x8
   b:   00 00 00 00
                        b: R_386_32 buf+0x4
   f:   48 8b 05 00 00 00 00    mov    0x0(%rip),%rax        # 16 <swap+0x16>
                        12: R_386_PC32       bufp0  -0x4
  16:   8b 00                   mov    (%rax),%eax
  18:   89 45 fc                mov    %eax,-0x4(%rbp)
  1b:   48 8b 05 00 00 00 00    mov    0x0(%rip),%rax        # 22 <swap+0x22>
                        1e: R_386_PC32       bufp0  -0x4
  22:   48 8b 15 00 00 00 00    mov    0x0(%rip),%rdx        # 29 <swap+0x29>
                        25: R_386_PC32       bufp1  -0x4
  29:   8b 12                   mov    (%rdx),%edx
  2b:   89 10                   mov    %edx,(%rax)
  2d:   48 8b 05 00 00 00 00    mov    0x0(%rip),%rax        # 34 <swap+0x34>
                        30: R_386_PC32       bufp1  -0x4
  34:   8b 55 fc                mov    -0x4(%rbp),%edx
  37:   89 10                   mov    %edx,(%rax)
  39:   5d                      pop    %rbp
  3a:   c3                      retq
```

(수행시의 PC 보정값)

Swap – Before Relocation

```
0000000000000000 <swap>:
   0:   55                          push   %rbp
   1:   48 89 e5                    mov    %rsp,%rbp
   4:   48 c7 05 00 00 00 00        movq   $0x0,0x0(%rip)        # f <swap+0xf>
                        7: R_386_PC32           bufp1  -0x8
   b:   00 00 00 00
                        b: R_386_32S buf+0x4
   f:   48 8b 05 00 00 00 00        mov    0x0(%rip),%rax        # 16 <swap+0x16>
```

Swap – After Relocation

```
0000000000400500 <swap>:
  400500:   55                          push   %rbp
  400501:   48 89 e5                    mov    %rsp,%rbp
  400504:   48 c7 05 ?? ?? ?? ??        movq   $0x60103c,????????(%rip) #600000 <bufp1>
  40050b:   3c 10 60 00
  40050f:   …                              (next instruction)
```

```
Disassembly of section .bss:

0000000000600000 <bufp1>:
```

PC-relative Address:

(3) movq를 수행할 시점의 PC는 ?

- movq 다음 instrution은 mov 0x0(%rip),%rax: 0040050f

(4) ?? ?? ?? ?? 에 들어갈 주소값을 구하시오.

addr(bup1) – (PC) => 00600000 – 0040050f = 001ffaf1 => f1 fa 1f 00