Initial count value

```
→    0x4004ef <main+8>         mov    DWORD PTR [rbp-0x4], 0x0
```

Loop will continue as long as current count value is 9 or lesser, aka < 10

```
→    0x400512 <main+43>        cmp    DWORD PTR [rbp-0x4], 0x9
     0x400516 <main+47>        jle    0x4004f8 <main+17>
```

Rsi : current count value

Rdi: message to be printed plus format string

```
→    0x4004f8 <main+17>        mov    eax, DWORD PTR [rbp-0x4]
     0x4004fb <main+20>        mov    esi, eax
     0x4004fd <main+22>        lea    rdi, [rip+0xa0]        # 0x4005a4
     0x400504 <main+29>        mov    eax, 0x0
     0x400509 <main+34>        call   0x4003f0 <printf@plt>
```

Increment count by 2

```
→    0x40050e <main+39>        add    DWORD PTR [rbp-0x4], 0x2
```

If current count value is 10 or 0xa

0xA is greater than 0x9, as such jump is not taken

```
gef➤  x/wx $rbp-4
0x7fffffffe41c: 0x0000000a
gef➤
```

```
     0x40050e <main+39>        add    DWORD PTR [rbp-0x4], 0x2
     0x400512 <main+43>        cmp    DWORD PTR [rbp-0x4], 0x9
→    0x400516 <main+47>        jle    0x4004f8 <main+17>        NOT taken [Reason: !(Z || S!=O)]
```

```
gef➤  print $eflags
$5 = [ IF ]
gef➤
```