Initial message

It prints “nested loops..” to the console

```
→   0x40057f <main+8>          lea    rdi, [rip+0xee]        # 0x400674
    0x400586 <main+15>         call   0x400470 <puts@plt>
```

Reference code(Outer loop)

```
 →    9       for(count=1;count<=3;count++)
```

Count=1

```
 →   0x40058b <main+20>          mov    DWORD PTR [rbp-0x4], 0x1
```

As long as count is lesser or equal to 3, (outer loop) continues

```
→   0x4005d2 <main+91>          cmp    DWORD PTR [rbp-0x4], 0x3
    0x4005d6 <main+95>          jle    0x400594 <main+29>
```

Reference code(Inner loop)

```
 for(output='A';output<='C';output++)
```

Output=’A’ as 0x41 is ‘A’

```
 →   0x400594 <main+29>          mov    BYTE PTR [rbp-0x5], 0x41
```

As long as the hex value of the char being compared is lesser or equals than C, (inner loop) continues

```
 →   0x4005be <main+71>          cmp    BYTE PTR [rbp-0x5], 0x43
     0x4005c2 <main+75>          jle    0x40059a <main+35>
```

Reference code

```
printf("Loop(%d) -> %c ", count, output);
```

Rdi – message to be printed and format specifier

Rsi – outer loop value

Rdx – inner loop value

```
    0x40059a <main+35>          movsx  edx, BYTE PTR [rbp-0x5]
    0x40059e <main+39>          mov    eax, DWORD PTR [rbp-0x4]
    0x4005a1 <main+42>          mov    esi, eax
→   0x4005a3 <main+44>          lea    rdi, [rip+0xda]        # 0x400684
    0x4005aa <main+51>          mov    eax, 0x0
    0x4005af <main+56>          call   0x400480 <printf@plt>
```

Memory to rax register, increment it by 1 and put it back to the original memory location.

In c it is equivalent to incrementing counter by 1.

```
 →   0x4005b4 <main+61>         movzx  eax, BYTE PTR [rbp-0x5]
     0x4005b8 <main+65>         add    eax, 0x1
     0x4005bb <main+68>         mov    BYTE PTR [rbp-0x5], al
```

If (inner loop) value is 0x44 which is greater than 0x43, loop will not continue.

```
gef➤  x/bx $rbp-5
0x7fffffffe41b: 0x44
```

```
    0x4005be <main+71>         cmp    BYTE PTR [rbp-0x5], 0x43
 →  0x4005c2 <main+75>         jle    0x40059a <main+35>        NOT taken [Reason: !(Z || S!=O)]
```

If (outer loop) value is greater 0x44, loop will not continue.

```
gef➤  x/wx $rbp-4
0x7fffffffe41c: 0x00000004
```

```
    0x4005d2 <main+91>         cmp    DWORD PTR [rbp-0x4], 0x3
 →  0x4005d6 <main+95>         jle    0x400594 <main+29>        NOT taken [Reason: !(Z || S!=O)]
```

Ending message

```
gef➤  x/s 0x400694
0x400694:       "\nAnd we are done!!"
```

```
 →   0x4005d8 <main+97>         lea    rdi, [rip+0xb5]        # 0x400694
     0x4005df <main+104>        call   0x400470 <puts@plt>
```