

## Ruby trunk - Bug #15731

### Wrong evaluation of many keyword default arguments in 2.3 - 2.5

03/27/2019 05:53 PM - marcandre (Marc-Andre Lafortune)

<b>Status:</b> Open	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> ruby 2.5.3p105	<b>Backport:</b> 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

#### Description

I don't know if it's worth fixing at this point, but we found a strange bug with evaluation of default keyword arguments when there are many of them (more than 32).

```
def foo(  
  k0: puts(0), k1: puts(1), k2: puts(2), k3: puts(3), k4: puts(4),  
  k5: puts(5), k6: puts(6), k7: puts(7), k8: puts(8), k9: puts(9),  
  k10: puts(10), k11: puts(11), k12: puts(12), k13: puts(13), k14: puts(14),  
  k15: puts(15), k16: puts(16), k17: puts(17), k18: puts(18), k19: puts(19),  
  k20: puts(20), k21: puts(21), k22: puts(22), k23: puts(23), k24: puts(24),  
  k25: puts(25), k26: puts(26), k27: puts(27), k28: puts(28), k29: puts(29),  
  k30: puts(30), k31: puts(31), k32: puts(32), k33: puts(33)  
)  
  k33  
end
```

```
puts "No params:"  
foo # Should print 1 to 33
```

```
puts "Only k33 param:"  
foo(k33: 1) # Should print 1 to 32
```

```
puts "Only k32 and k33 params:"  
r = foo(k32: 1, k33: 1) # Should print 1 to 31 and return 1
```

```
puts "Result: #{r.inspect}"
```

#### Ruby 2.4:

last case is wrong. It prints 1 to 33 instead of 1 to 31 and returns nil instead of 1.

#### Ruby 2.5:

same result for last case

first two cases evaluates the default exactly the wrong parameters: those that are given and not for those not given. So it prints nothing and 33 respectively, instead of 1 to 33 and 1 to 32!

#### Ruby 2.6:

results are ok

This strange behavior disappears with fewer keyword arguments.

#### History

#1 - 03/27/2019 06:37 PM - marcandre (Marc-Andre Lafortune)

Edits:

Print 1 to ... => Print 0 to ...

Ruby 2.4 bug => Ruby 2.2/2.3/24 bug

Ruby 2.6 ok => Ruby 2.1 and 2.6 ok

We => Me and Maxime Lapointe