INF600A-20 Quiz formatif no. 4
30 octobre 2018

Indiquez ce qui sera affiché par `irb` pour chaque expression.

1. Utilisations de `yield`

```ruby
def foo( n )
  n.times { yield }
end

def bar( elems )
  k = 1
  elems.each do |e|
    yield( k * e )

    k += 1
  end
end
```

```ruby
def baz( x = 0 )
  return x unless block_given?

  yield( yield(x) )
end

def qux( x = 0, y: nil )
  return yield( x, 1 ) unless y

  yield( x, y )
end
```

```ruby
# a.
>> x = 0; foo(10) { x += 2 }; x
=>
```

```ruby
# b.
>> bar([10, 20, 30]) { |k| puts k }

=>
```

```ruby
# c.
>> baz(1)
=>

>> baz { |x| x + 5 }
=>

>> baz("12") { |x| x * 2 }
=>
```

```ruby
# d.
>> qux { |a, b| a + b }
=>

>> qux(10) { |a, b| a * b }
=>

>> qux("5", y: 3) { |a, b| a * b }
=>
```

2. Opérations de style fonctionnel et module `Enumerable`

Soit l'extension suivante de la classe `Fixnum` (nombres entiers) :

```
class Fixnum
  include Enumerable

  def each # Enumere les chiffres du nombre, de droite a gauche.
    (yield(0); return) if self == 0
    v = self
    while v != 0; yield(v % 10); v /= 10; end
  end
end
```

Exemple d'utilisation : `123.map { |x| x } == [3, 2, 1]`

```
# a.
>> 8703.select { |x| x >= 4 }
=>

# b.
>> 8703.reject { |x| x.even? }
=>

# c.
>> 8703.reject { |x| x }
=>

# d.
>> 8703.map { |x| x * x }
=>

# e.
>> 8703.map { |x| x.include? 3 }
=>

# f.
>> 8703.map { |x| x.map { |y| y + 1 } }
=>

# g.
>> 8703.reduce(0) { |a, x| a + 2 * x }
=>

# h.
>> 8703.reduce([[], []]) { |a, x| (x.even? ? a[0] : a[1]) << x; a }
=>
```