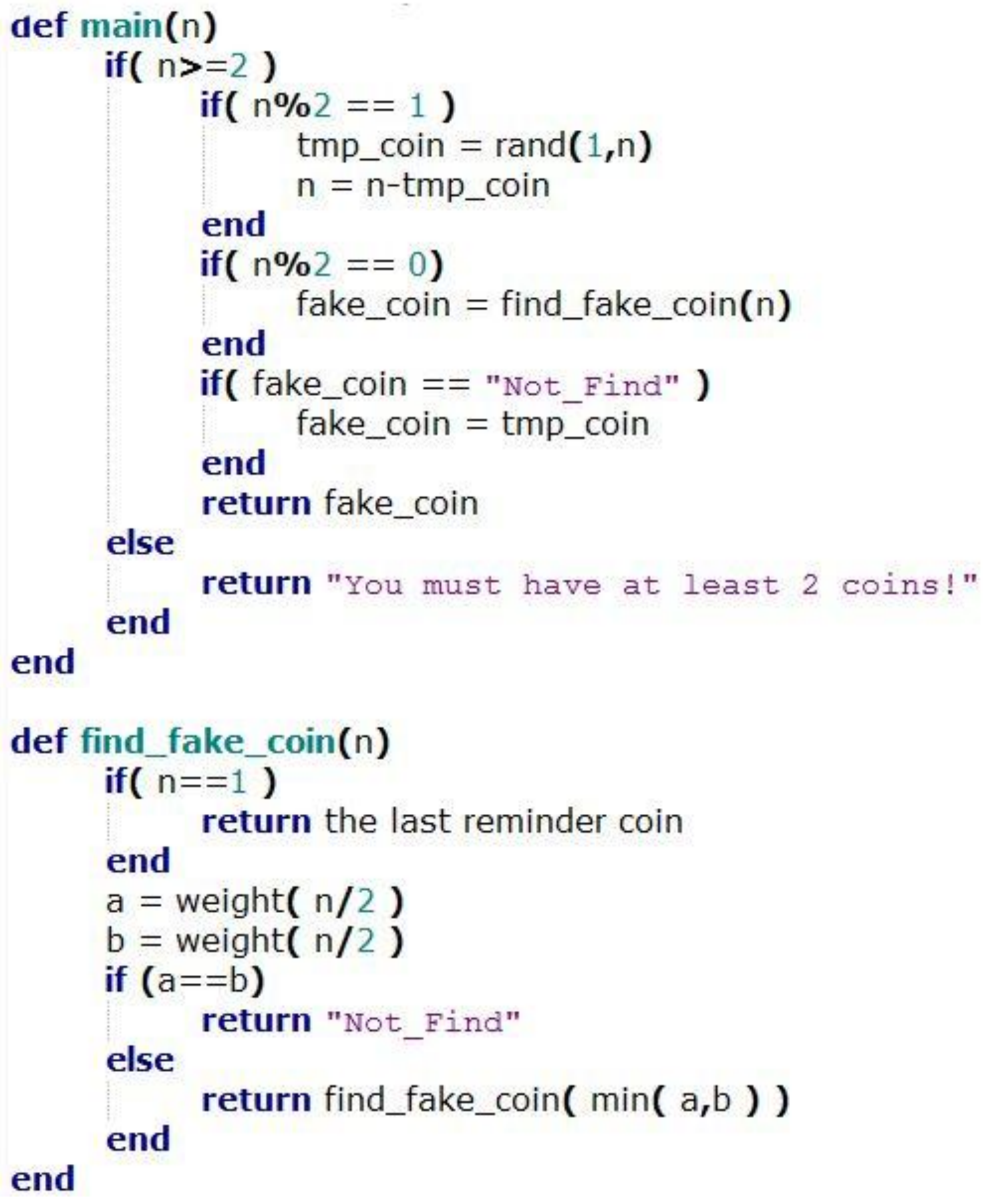
The Fake-Coin Problem: Decrease by a Constant Factor

- Problem:
  - Among *n* identical looking coins, one is a fake (and weighs less)
  - We have a balance scale which can compare any two sets of coins
- Algorithm:
  - Divide into two size $\lfloor n/2 \rfloor$ piles (keeping a coin aside if *n* is odd)
  - If they weigh the same then the extra coin is fake
  - Otherwise proceed recursively with the lighter pile
- Efficiency:
  - $W(n) = W(\lfloor n/2 \rfloor) + 1$ for $n > 1$
  - $W(n) = \lfloor \log_2 n \rfloor = \Theta(\log_2 n)$
- But there is a better ($\log_3 n$) algorithm



```
def main(n)
    if( n>=2 )
        if( n%2 == 1 )
            tmp_coin = rand(1,n)
            n = n-tmp_coin
        end
        if( n%2 == 0)
            fake_coin = find_fake_coin(n)
        end
        if( fake_coin == "Not_Find" )
            fake_coin = tmp_coin
        end
        return fake_coin
    else
        return "You must have at least 2 coins!"
    end
end

def find_fake_coin(n)
    if( n==1 )
        return the last reminder coin
    end
    a = weight( n/2 )
    b = weight( n/2 )
    if (a==b)
        return "Not_Find"
    else
        return find_fake_coin( min( a,b ) )
    end
end
```