

1. Escreva uma função que recebe um array e imprime todos os elementos que há nele. Considere o uso da função `table.unpack` e avalie vantagens e desvantagens do uso dela para a implementação solicitada.
2. Escreva uma função `fibs` que retorna um *closure iterator* que produz os primeiros `n` números da sequência de Fibonacci. Por exemplo,

```
s = 0
for i in fibs(5) do
    s = s + i
end
print(s) -- 12
```

3. A partir da execução do código Lua abaixo, defina a sequência do que será impresso.

```
function f1 (a)
    print("f1", a)
    coroutine.yield(2*a)
    f2(a)
    return a*7
end
function f2 (a)
    print("f2", a)
    return coroutine.yield(3*a)
end

co1 = coroutine.create(function (a,b)
    print("co1", a, b)
    local r = f1(a+1)
    print(r)
    print(a,b)
    local r, s = coroutine.yield(a+b, a-b)
    print ("co1", coroutine.resume(co2,a+1,b+1))
    return b, "fim_co1"
end)

co2 = coroutine.create(function (a,b)
    print("co2", a, b)
    print ("co2", coroutine.resume(co3,a+1,b+1))
    return b, "fim_co2"
end)

co3 = coroutine.create(function (a,b)
    print("co3", a, b)
    coroutine.yield(b*57)
    local r, s = coroutine.yield(a+b, a-b)
    print("co3", r, s)
    print("co3", a+1, b+1)
```

```
    end)

print("main", coroutine.resume(co1, 6, 14))
print("main", coroutine.resume(co1))
print("main", coroutine.resume(co1, "x", "y"))
```