P1 de ICC

1. (3,0) Escreva uma função que receba uma lista de valores de rendimentos e um número n, e retorne uma lista com os n maiores impostos correspondentes a esses rendimentos. Assuma que já existe 1) uma função `ordena`, que dada uma lista de números retorna uma lista com os mesmos números ordenados em ordem decrescente; e 2) uma função `imposto`, que dado um valor de rendimento retorne o total de imposto a pagar. (DICA: lembre-se que os n maiores impostos são os n primeiros elementos da lista de impostos ordenada em ordem decrescente.)

2. (3,0) O predicado abaixo determina se um dado ano é bissexto:

```
(define divide (lambda (a b) (zero? (remainder a b))))

(define bissexto
  (lambda (ano)
    (or (divide ano 400)
        (and (divide ano 4) (not (divide ano 100))))))
```

   (a) Reescreva o predicado `bissexto` acima, sem usar `and`, `or` nem `not`. Use `ifs` e/ou `conds`.

   (b) Escreva um predicado que receba uma lista de anos e verifique se todos eles são bissextos (isso é, retorna `#t` somente se todos os anos da lista forem bissextos).

3. (3,0) Usando recursão final, escreva um predicado que receba uma lista de números inteiros e retorne `#t` somente se mais da metade dos números na lista forem pares. Lembre-se que o predicado pré-definido `even?` testa se um número é par. (DICA: use dois acumuladores.)