P2 de ICC

Podemos representar uma conta bancária, de forma muito simplificada, como uma lista de números, correspondendo aos vários lançamentos de um determinado período, em ordem cronológica; créditos são representados por números positivos, e débitos por números negativos. Por exemplo, a lista

```
(130.00 -90.00 -10.00)
```

representa uma conta com um saldo inicial de R$130,00, e que teve em seguida um débito de R$90,00 e depois outro de R$10,00. Observe que o saldo de uma conta é dado pelo somatório dos elementos da lista; na conta acima, o saldo final seria de R$30,00.

1. Usando `do`, escreva uma função `saldo` que receba uma lista representando uma conta e retorne o saldo dessa conta.

2. Escreva uma função que receba dois conjuntos, representados como listas, e retorne a interseção desses conjuntos.

3. Considere agora uma lista associativa, que associa CPFs a contas bancárias; por exemplo:

   ```
   (define CONTAS '(
     (756987001-43   (130.00 -54.10 -12.00 -10.00 100.00))
     (065999103-90   (1000.00 -230.00 -500.00 433.30))
     (000132871-44   (100.00 -20.00 50.00 -400.00))
     (877569233-14   (100.00 100.00 100.00 -500.00 20.00))
      ...))
   ```

   Escreva uma função que receba uma lista como `CONTAS` e um número de CPF de um correntista, e retorne o saldo desse correntista.

4. Escreva outra função, que receba uma lista associativa como descrita acima, e retorne o conjunto dos CPFs dos correntistas que estão com saldo negativo, representado como uma lista. Por exemplo,

   `(vermelho CONTAS)` $\rightsquigarrow$ `(000132871-44 877569233-14)`

5. Imagine que o banco tenha um conjunto de contas consideradas especiais. Esse conjunto é representado por uma lista de CPFs; por exemplo:

   ```
   (define ESPECIAL '(065999103-90 77100067-23 ...))
   ```

   Baseado nas estruturas `CONTAS` e `ESPECIAL`, escreva uma função que retorne os CPFs dos correntistas especiais que estão com saldo negativo.

Atenção 1: sempre que necessário, use uma função passada em uma questão na solução de outra questão, mesmo sem ter escrito a função.

Atenção 2: Qualquer outra função não primitiva que for usada deverá ser definida (as primitivas são `car`, `cdr`, `cons`, `caar`, etc).