# Folha TP 4



Nota. Sempre que for pedido que implemente uma função (problemas 1, 2, 4, 5, 6, 8 e 9), deve implementar uma função pura; em particular, dentro da definição da função: (i) não deve haver instruções de entrada ou saída, (ii) deve ser sempre retornado um resultado.
Nota. Os exercícios 3, 4 e 5 já foram resolvidos em aulas anteriores.

```
----------------------
1)  Escreva uma função ´comuns(s,t)´ que retorna uma lista das letras
    minúsculas comuns às strings ´s´ e ´t´, sem repetição.
    Exemplo: comuns("Vacas", "Vatatas") -> ["a","s"]
    Note que "V" não ocorre na lista e que "a" ocorre uma só vez.

2)  Considera-se para este problema que uma string ´s´ representa um
    valor inteiro ou decimal, se for constituída
       - por uma sequência de 1 ou mais dígitos ... ou ...
       - por uma sequência de 1 ou mais dígitos, seguida de ´.´,
         seguida de uma sequência de 0 ou mais dígitos.
    a) Escreva uma função ´numero(s)´ que retorna ´True´ se ´s´ representa
       um valor inteiro ou decimal e ´False´ caso contrário.
       Exemplos: numero("123") -> True, numero(".123") -> False,
                 numero("12.") -> True, numero("1e2")  -> False
    b) Escreva uma função ´valor(s)´ que retorna o valor (float)
       representado por ´s´ (se numero(s)==True) ou -1
       (se numero(s)==False)
       Exemplos: valor("123") -> 123.0, valor(".123") -> -1,
                 valor("12.") -> 12.0, valor("1e2")   -> -1

3) Considere a seguinte função (não pura)
       def base(n,b):
           while n>0:
               print n%b,
               n=n/b
   onde n é um inteiro não negativo e b um inteiro maior ou igual a 2.
   a) Descreva, usando no máximo 2 linhas de texto, o efeito da função.
   b) Qual o efeito das seguintes chamadas?

       i  - >>> base(12,2)
       ii - >>> base(35,20)

4) Utilizando listas, pretende-se implementar uma função

                            converte(n,b)

   que converte o inteiro n para a base b, ficando o resultado na
   lista que é retornada.

         converte(14,2)  ->  [1,1,1,0]   (pois 14 = 8+4+2)
         converte(47,16) ->  [2,15]      (pois 47 = 2*16 + 15)
```

Sugestão: aproveite a ideia do problema 1; use uma lista r, inicialmente nula (r=[]), acrescentando cada dígito d à esquerda (porquê à esquerda?) com r=[d]+r; no fim, r é retornada.

Ex: converte(14,2)

| n | d | r |
|---|---|---|
| | | [] |
| 14 | 0 | [0] |
| 7 | 1 | [1,0] |
| 3 | 1 | [1,1,0] |
| 1 | 1 | [1,1,1,0] |
| 0 | | |

5) Implemente uma função "valor(li,b) -> int" que implementa a operação inversa (ver problema anterior), isto é, converte uma representação numa base para um inteiro.

Ex: valor([1,1,0,0],2) -> 12

Não pode usar a operação de potênciação (**) nem listas auxiliares.

6) Implemente uma função "convertex(n,b)" com o mesmo objectivo do problema 2, mas usando o seguinte método

- numa primeira fase obtém-se a lista r invertida (usando a operação r=r+[d] (mais eficiente, em princípio, que r=[d]+r)
  Ex: n=14, b=2, r=[0,1,1,1]
- numa segunda fase, inverte-se a lista r
  Ex [0,1,1,1] -> [1,1,1,0]

7) Explique em que sentido podemos dizer que as funções ´converte(n,b)´ e ´valor(li,b)´ são inversas uma da outra.

8) Supondo que 2<=b<=16 e que os dígitos nas bases até 16 são representados por

"0", "1", "2", "3", "4", "5", "6", "7",
"8", "9", "a", "b", "c", "d", "e", "f"

escreva uma função conv(n,b) que converta n para a base b, usando esta notação.

Ex: conv(254,16) -> ['f', 'e']

Sugestão: use a string auxiliar "0123456789abcdef"

9) Um resultado mais agradável que o obtido no problema anterior pode-se conseguir usando um string r, fazendo-se inicialmente r="" e adicionando-se cada dígito d (string) com a operação r=d+r
Implemente uma função convs(n,b) baseada nesta ideia.

Exs:

conv1(254,16) -> "fe"
conv1(1888,10) -> "1888"