

1) Escreva uma função

```
coloca(n,i,lista)
```

onde 'lista' é uma lista de inteiros e que tem como efeito inserir o valor 'n' na posição correspondente ao índice 'i' da lista 'lista'.

O valor retornado deve ser 'lista' (que entretanto foi transformada; note que a função não é pura).

Exemplo: `coloca(5,2,[8,3,4,6,9])` -> `[8,3,5,4,6,9]`
(coloca 5 na posição de índice 2 da lista `[8,3,4,6,9]`)

Na seguinte sugestão usa-se o exemplo em cima.

Nota. "_" indica que o conteúdo é irrelevante

- a) Criar um novo espaço no fim `[8,3,4,6,9]` -> `[8,3,4,6,9,_]`
- b) "Chega-se à frente" a parte apropriada:
`[8,3,4,6,9,_]` -> `[8,3,_,4,6,9]`
- c) Colocar 'n' na posição 'i'. `[8,3,_,4,6,9]` -> `[8,3,5,4,6,9]`

Notas.

- As funções do python que pode utilizar são apenas: 'len' e 'append'; não pode por exemplo, usar 'del', 'insert'...
- Existe já em python o método 'insert' que é equivalente a 'coloca' no sentido em que
'coloca(n,i,lista)' é equivalente a 'lista.insert(i,n)'

2) Escreva uma função pura

```
merge(a,b)
```

onde 'a' e 'b' são listas de inteiros, ordenadas por ordem crescente (ou, mais rigorosamente, "não decrescente").
A função deve retornar uma nova lista (também ordenada por ordem crescente) com todos os elementos de 'a' e de 'b'.

Exemplo:

```
merge([2,3,5,8],[1,2,2,4,8,10,20]) ->  
[1,2,2,2,3,4,5,8,8,10,20]
```

Sugestão: Use 2 índices i, e j, um para cada lista, inicializados em 0. A lista resultado dependerá essencialmente de comparações de `a[i]` com `b[j]`.

3) Escreva uma função

`compara(s,t)`

que compara lexicograficamente as strings 's' e 't'.
Supõe-se que 's' e 't' só contêm letras minúsculas não acentuadas. A função deve retornar

-1 se $s < t$
0 se $s = t$
1 se $s > t$

Exemplos:

`compara("ai","baca") -> -1` (1)
`compara("ai","ai") -> 0` (2)
`compara("va","vaca") -> -1` (3)
`compara("", "a") -> -1` (4)

Seja $ns = \text{len}(s)$, $nt = \text{len}(t)$. Regras da comparação, supondo-se a ordenação habitual das letras $"a" < "b" < \dots < "z"$.

Método a usar: Percorrer "s" e "t" enquanto as letras correspondentes (na mesma posição) forem iguais e nenhuma das strings se tiver "esgotado".

Se 's' e 't' concordam até ao índice 'i-1':

- a) se $i < ns$ e $i < nt$: (nenhuma das strings se esgotou):
 - $s[i] < t[i] \rightarrow -1$ (exemplo (1))
 - $s[i] > t[i] \rightarrow 1$
- b) se $i = ns$ e $i < nt$: (esgotou-se 's')
 - $\rightarrow -1$ (exemplos (3) e (4))
- c) se $i < ns$ e $i = nt$: (esgotou-se 't')
 - $\rightarrow 1$
- d) se $i = ns$ e $i = nt$: (ambas as strings se esgotaram)
 - $\rightarrow 0$ (exemplo (2))

Nota. Já existe implementada no python uma comparação que usa exactamente estas regras. Basta fazer, por exemplo, $s < t$. Mas muitas vezes nos problemas que ocorrem na prática, este tipo de comparação não é adequado sendo necessário implementar uma função de comparação específica.