Capítulo 6

Funções revisitadas

6.1 Funções recursivas

Nota: Nos exercícios desta aula não pode utilizar a atribuição nem os ciclos `while` e `for`.

1. Escreva a função recursiva `apenas_digitos_impares` que recebe um número inteiro não negativo n, e devolve um inteiro composto apenas pelos dígitos ímpares de n. Se n não tiver dígitos ímpares, a função deve devolver zero. Não pode usar cadeias de caracteres. Por exemplo,

   ```
   >>> apenas_digitos_impares(468)
   0
   >>> apenas_digitos_impares(12426374856)
   1375
   ```

2. Escreva a função recursiva `junta_ordenadas` que recebe como argumentos duas listas ordenadas por ordem crescente e devolve uma lista também ordenada com os elementos das duas listas. Não é necessário validar os argumentos da sua função. Por exemplo,

   ```
   junta_ordenadas([2, 5, 90], [3, 5, 6, 12])
   [2, 3, 5, 5, 6, 12, 90]
   ```

3. Escreva a função recursiva `sublistas` que recebe uma lista, e tem como valor o número total de sublistas que esta contém. Por exemplo,

   ```
   >>> sublistas([[1], 2, [3]])
   2
   >>> sublistas([[[[[1]]]]])
   4
   >>>sublistas(['a', [2, 3, [[[1]], 6, 7], 'b']])
   4
   ```

4. Escreva a função recursiva `soma_n_vezes` que recebe três números inteiros, a, b e n, e que devolve o valor de somar n vezes a a b, ou seja, $b + a + a + \ldots + a$, n vezes. A sua função não pode usar a operação de multiplicação. Por exemplo,

```
>>> soma_n_vezes(3, 2, 5)
17
```

5. Escreva a função recursiva `soma_els_atomicos` que recebe como argumento um tuplo, cujos elementos podem ser outros tuplos, e que devolve a soma dos elementos correspondentes a tipos elementares de dados que existem no tuplo original. Não é necessário verificar os dados de entrada. Por exemplo,

```
>>> soma_els_atomicos((3, ((((((6, (7, ))), ), ), ), ), 2, 1))
19
>>> soma_els_atomicos(((((),),),))
0
```

6. Escreva a função recursiva `inverte` que recebe uma lista e devolve a lista invertida. Por exemplo,

```
>>> inverte([3, 4, 7, 9])
[9, 7, 4, 3]
```

7. Suponha que a operação `in` não existia em Python. Escreva a função recursiva `pertence` que recebe uma lista e um elemento e devolve `True` se o elemento pertence à lista e `False` em caso contrário. Por exemplo,

```
>>> pertence([3, 4, 5], 2)
False
>>> pertence([3, 4, 5], 5)
True
```

8. Escreva a função recursiva `subtrai` que recebe duas listas e devolve a lista que corresponde a remover da primeira lista todos os elementos que pertencem à segunda lista. Por exemplo,

```
subtrai([2, 3, 4, 5], [2, 3])
[4, 5]
subtrai([2, 3, 4, 5], [6, 7])
[2, 3, 4, 5]
```

9. Escreva a função recursiva `parte` que recebe uma lista de números e um número e que devolve uma lista de duas listas, a primeira lista contém os elementos da lista original menores que o número dado (pela mesma

ordem) e a segunda lista contém os elementos da lista original maiores ou iguais que o número dado (pela mesma ordem). Não é necessário verificar a coreção dos dados de entrada. **Sugestão**: Use uma função auxiliar. Por exemplo,

```
>>> parte([3, 5, 1, 4, 5, 8, 9], 4)
[[3, 1], [5, 4, 5, 8, 9]]
```

10. Escreva a função recursiva `maior` que recebe uma lista de números e devolve o maior número da lista. **Sugestão**: Use uma função auxiliar. Por exemplo,

```
>>> maior([5, 3, 8, 1, 9, 2])
9
```