

Exercícios extra para revisão

Exercícios

1. A sequência de Fibonacci começa com os números 0 e 1. Depois, cada elemento consecutivo da sequência obtém-se pela soma dos dois elementos anteriores. No programa `genFibonacci.py`, complete a função `genFibonacci(n)` para devolver uma lista com os n primeiros números de Fibonacci. Por exemplo, se $n=6$, deve devolver `[0, 1, 1, 2, 3, 5]`. A função só tem de funcionar para $n \geq 2$. Repare que o programa já faz várias invocações e verifica os resultados com instruções `assert`. Ao executar, se surgir um `AssertionError`, significa que a função está errada. (Ou faça no [CodeCheck](#).)
2. Escreva uma função que verifique se uma string contém caracteres consecutivos iguais. Por exemplo, `containsDoubles("pool")` deve devolver `True`. Faça a função no programa `checkDoubles.py` e verifique que passa os testes. Acrescente mais alguns testes que lhe pareçam adequados. Lembre-se que deve testar *casos extremos*.
3. O ficheiro `wordlist.txt` contém uma lista de palavras da língua inglesa, com uma palavra por linha. No programa `checkDoubles.py`, complete a função `findLinesWithDoubles` para descobrir as palavras com letras duplicadas num ficheiro como esse.
4. Imagine que está a fazer palavras cruzadas (em Inglês) e falta-lhe uma palavra com o padrão `"?YS???Y"`, onde os `"?"` representam letras por preencher. Complete o programa `findWords.py` para o ajudar a descobrir a palavra. O programa já inclui instruções para ler uma lista de palavras de língua inglesa a partir do ficheiro `wordlist.txt`. Faça a função `filterPattern(lst, pattern)` para extrair duma lista de strings as strings que têm o padrão dado. *Sugestão: invoque a função `matchesPattern` para testar cada palavra.* Complete a função `matchesPattern(s, pattern)` para devolver `True` se `s` corresponder ao padrão fornecido e `False`, no caso contrário. Uma string `s` corresponde ao padrão se e só se tiver os mesmos caracteres que o padrão nas mesmas posições, exceto onde o padrão tem `?`. Nas posições dos `?`, não importa que caracteres estão na string `s`. A correspondência não deve fazer distinção entre maiúsculas e minúsculas. ([CodeCheck](#).)

5. A função `main` no programa `imctable.py` define uma lista com pesos e alturas de várias pessoas e invoca diversas funções para processar e mostrar esses dados. Analise o programa e complete as funções que faltam para que funcione corretamente. (Pode fazer no [CodeCheck](#).)
- a) A função `printTable` deve imprimir uma tabela com 4 colunas: nome, peso, altura e índice de massa corporal (IMC). O IMC pode ser calculado por $\text{peso}/\text{altura}^2$. As colunas numéricas devem aparecer ajustadas à direita e com um número fixo de casas decimais, como no exemplo abaixo.
 - b) A função `inputBetween` deve pedir e devolver um valor introduzido pelo utilizador, mas apenas se estiver dentro dos limites indicados. Senão, deve avisar e voltar a pedir o valor até que a condição se verifique.
 - c) A função `selectTaller` deve devolver uma lista com os registos das pessoas mais altas que o limite dado.

O exemplo abaixo demonstra o funcionamento pretendido do programa.

```
extra1$ python3 imctable.py
Nome          Peso  Altura   IMC
John          64    1.76    20.9
Berta         64    1.61    24.6
Maria         45    1.72    15.3
Andy          98    1.81    30.0
Lisa          47    1.62    17.8
Kelly         83    1.78    26.3

altura minima? 3
Value must be in [1.1, 2.5]!
altura minima? 1
Value must be in [1.1, 2.5]!
altura minima? 1.7
Nome          Peso  Altura   IMC
John          64    1.76    20.9
Maria         45    1.72    15.3
Andy          98    1.81    30.0
Kelly         83    1.78    26.3
```