

Lista de Exercícios 2

Dicionário

1. Escreva uma função que conta a quantidade de cada vogal em um texto e armazena tal quantidade em um dicionário, onde a chave é a vogal considerada. O texto deve ser modificado para ter todas as letras maiúsculas.
2. Escreva um programa que lê duas notas de vários alunos e armazena tais notas em um dicionário, onde a chave é o nome do aluno. A entrada de dados deve terminar quando for lida uma string vazia como nome. Escreva uma função que retorna a média do aluno, dado seu nome.
- 3.a) Escreva a função decodifica para exibir na tela a decodificação de mensagens. As mensagens foram codificadas utilizando símbolos especiais no lugar das vogais. Para decodificar a mensagem deve ser utilizado o seguinte dicionário dVogal: dVogal = {'@':'A','&':'E','%':'I','*':'O','\$':'U'}
A função decodifica:
 - recebe uma string com a mensagem codificada em maiúsculas e o dicionário dVogal
 - retorna uma string com a mensagem decodificada.EXEMPLOS:
A mensagem codificada 'B*N&C@' retorna 'BONECA'
A mensagem codificada 'Q\$@ND* @ CH\$V@ P@SS*\$' retorna 'QUANDO A CHUVA PASSOU'
- 3.b) Teste a função decodifica para as mensagens abaixo (msg1 e msg2) exibindo as respostas retornadas.
msg1: 'Q\$@ND* @ CH\$V@ P@SS*\$'
msg2: '*\$R* BR@NC* *\$R* N&GR*'
Obs: dicionários devem ser manipulados como dicionários. Não é para transformar em outra estrutura (como lista) e usar a outra estrutura.
4. Escreva uma função que recebe uma lista de compras e um dicionário contendo o preço de cada produto disponível em uma determinada loja, e retorna o valor total dos itens da lista que estejam disponíveis nesta loja. Por exemplo, para os dados:
lista_de_compras = ['biscoito', 'chocolate', 'farinha']
Supermercado = {'amaciante':4.99,'arroz':10.90,'biscoito':1.69,
 'cafe':6.98, 'chocolate':3.79,'farinha':2.99}

5. Afinidades entre pessoas podem ser descritas através de dicionários. Por exemplo, se tivermos:

- Leo gosta de Sofia,
- Marcos gosta de Andrea,
- Sofia gosta de Leo,
- Alex gosta de Andrea, e
- Andrea gosta de Marcos

Podemos representar estas relações através do dicionário:

afinidades = { 'Leo': 'Sofia', 'Marcos': 'Andrea', 'Sofia': 'Leo',
 'Alex': 'Andrea', 'Andrea': 'Marcos' }

Escreva uma função que recebe um dicionário de afinidades e retorna uma lista de pares com afinidade mútua, ou seja, onde um gosta do outro. Por exemplo, para o dicionário acima, a função deve retornar:

[('Marcos', 'Andrea'), ('Sofia', 'Leo')]

6. O Zodíaco chinês é composto por animais com ciclo de 12 anos. Uma maneira simplificada de encontrá-lo é verificar apenas o ano de nascimento da pessoa do seguinte modo:

ano do nascimento % 12	Signo
0	Macaco
1	Galo
2	Cão
3	Porco
4	Rato
5	Boi
6	Tigre
7	Coelho
8	Dragão
9	Serpente
10	Cavalo
11	Carneiro

- a) Crie um dicionário com os signos
- b) Crie um dicionário com o nome e a data de nascimento dos membros de sua família no formato dd/mm/aaaa
- c) Faça uma função que, usando os dicionários nos itens a e b, mostre o signo de cada membro de sua família

7. Escreva um programa para armazenar uma agenda de telefones em um dicionário. Cada pessoa pode ter um ou mais telefones e a chave do dicionário é o nome da pessoa. Seu programa deve ter as seguintes funções: **incluirNovoNome** – essa função acrescenta um novo nome na agenda, com um ou mais telefones. Ela deve receber como argumentos o nome e os telefones.

incluirTelefone – essa função acrescenta um telefone em um nome existente na agenda. Caso o nome não exista na agenda, você deve perguntar se a pessoa deseja ser incluída. Caso a resposta seja afirmativa, use a função anterior para incluir o novo nome.

excluirTelefone – essa função exclui um telefone de uma pessoa que já está na agenda. Se a pessoa tiver apenas um telefone, ela deve ser excluída da agenda.

excluirNome – essa função exclui uma pessoa da agenda.

consultarTelefone – essa função retorna os telefones de uma pessoa na agenda.

Considere a existencia de outra agenda. Pode haver nomes comuns às agendas. Faça a união das agendas. Considere duas possibilidades:

- quando houver coincidência de nomes, os valores da segunda agenda prevalecem.
- quando houver coincidência de nomes, acrescente os valores da segunda que não constam na primeira.

8. Escreva uma função que receba uma lista e retorne uma nova lista sem elementos duplicados. Lembre que os elementos duplicados não precisam aparecer em posições consecutivas. Dica: use um dicionário

9. Escreva uma função que converte números inteiros entre 1 e 999 para algarismos romanos. Não converta o número para uma string.

Use os três dicionários abaixo:

- UNIDADES = { 0: '', 1: 'I', 2: 'II', 3: 'III', 4: 'IV', 5: 'V', 6: 'VI', 7: 'VII', 8: 'VIII', 9: 'IX' }
- DEZENAS = { 0: '', 1: 'X', 2: 'XX', 3: 'XXX', 4: 'XL', 5: 'L', 6: 'LX', 7: 'LXX', 8: 'LXXX', 9: 'XC' }
- CENTENAS = { 0: '', 1: 'C', 2: 'CC', 3: 'CCC', 4: 'CD', 5: 'D', 6: 'DC', 7: 'DCC', 8: 'DCCC', 9: 'CM' }

10. Para controlar a circulação de carros e motos nas ruas das cidades, algum órgão governamental associa uma placa a cada veículo. Para emitir uma multa ou denunciar um roubo à polícia, informa-se a placa que identifica o veículo. A placa é formada de letras e números e funciona como uma chave para encontrar o veículo desejado.

Faça um programa que leia as seguintes informações das infrações registradas no último ano: placa do carro, local, grau da infração, descrição da infração. O programa deve exibir as placas dos carros e respectivas infrações dos motoristas que devem ter a carteira suspensa (soma dos pontos ≥ 20)

Dica: crie também um dicionário para a pontuação por grau de infração.

11. Construa a função `freqPalavras` que receba uma string e retorne um dicionário onde cada palavra dessa string é uma chave e que tenha como valor o número de vezes que a palavra ocorre.

Por exemplo:

- `freqpalavras("dinheiro é dinheiro e vice versa")`
- retorna o dicionário: `{"dinheiro":2, "é": 1, "e": 1, "vice": 1, "versa":1}`

12. Uma pessoa pode ser representada como um ponto em um plano n-dimensional, onde cada eixo representa uma preferência avaliada: número inteiro de 0 a 10. Por exemplo, se a Sofia gosta muito de viajar (9) e pouco de ler (2) e o Pedro gosta pouco de viajar (2) e muito de ler (9) poderiam ser representados por:



A distância Euclidiana entre estes pontos pode ser usada como uma medida de similaridade entre eles.

Faça uma função que receba o nome e as notas dadas a cada um dos 6 aspectos avaliados por pessoas que inscreveram-se em um site de relacionamento. Esta função deve permitir que alguém, ao fornecer seu nome, receba o nome de 3 pessoas com que tenha afinidade (distância < 1.0). Caso esta pessoa ainda não esteja no cadastro, ela deve responder ao questionário e ser incluída.

13. Considere o dicionário abaixo com os produtos por mercado:

```
dProdsDoMerc= { 'Qbarato': {'biscoito': 4.3, 'leite': 3.2, 'suco': 7.1, 'chocolate': 6.4, 'detergente': 3.2, 'cerveja': 6.4, 'manteiga': 8.7},  
                 'UltraK': {'biscoito': 3.5, 'leite': 3.3, 'suco': 8.9, 'chocolate': 6.9, 'detergente': 4.2, 'cerveja': 6.4, 'manteiga': 8.7},
```

```
'Market': {'biscoito': 4.5, 'leite': 3.2, 'suco': 7.5, 'chocolate': 6.6, 'detergente': 3.8,
'cerveja': 6.5, 'manteiga': 9.2},
'Preferido': {'biscoito': 4.65, 'leite': 3.4, 'suco': 8.1, 'chocolate': 8.1, 'detergente': 3.3,
'cerveja': 6.5, 'manteiga': 8.9},
'Escolhido': {'biscoito': 5.2, 'leite': 3.3, 'suco': 8.3, 'chocolate': 7.5, 'detergente': 3.9,
'cerveja': 6.4, 'manteiga': 8.6}
}
```

Este dicionário é um dicionário de dicionários onde cada item é:

SUPERMERCADO : { dicionário_de_Preços_Por_Produto nesse supermercado }

Obs1: Todos os mercados têm todos os produtos.

Obs2: dicionários devem ser manipulados como dicionários. Não é para transformar para outra estrutura (como lista) e usar a outra estrutura.

Escreva a função totalDaCompra que:

- recebe o dicionário de produtos por mercado dProdsDoMerc descrito, o nome de um supermercado e um outro dicionário que representa a lista de compras de uma pessoa. Esse dicionário (a lista de compras) tem as seguintes informações:

dCompras = { PRODUTO1: QUANT1, PRODUTO2:QUANT2, PRODUTO3: QUANT3 ...}

- retorna o valor total da compra nesse supermercado.

Obs1: quantidades são sempre um número inteiro

Obs2: É garantido que o mercado possui todos os produtos do dicionário de compra.

EXEMPLO: A função retornaria 34.2, caso recebesse o dicionário apresentado, o nome do supermercado 'Qbarato' e o dicionário de compras = {'biscoito': 2, 'leite': 6, 'chocolate': 1}

Escreva a função criaDicMercadosDoProd que:

- recebe o dicionário de produtos por mercado dProdsDoMerc

- retorna o dicionário com os mercados por produto. O dicionário de mercados do produto é um dicionário de dicionários onde cada item é:

PRODUTO: { dicionário_de_Preços_Por_Mercado desse PRODUTO }

EXEMPLO: Se o dicionário recebido pela função fosse o dicionário apresentado, seria retornado:

```
{ 'biscoito': {'Qbarato': 4.3, 'UltraK': 3.5, 'Market': 4.5, 'Preferido': 4.65, 'Escolhido': 5.2},
'leite': {'Qbarato': 3.2, 'UltraK': 3.3, 'Market': 3.2, 'Preferido': 3.4, 'Escolhido': 3.3},
'suco': {'Qbarato': 7.1, 'UltraK': 8.9, 'Market': 7.5, 'Preferido': 8.1, 'Escolhido': 8.3},
'chocolate': {'Qbarato': 6.4, 'UltraK': 6.9, 'Market': 6.6, 'Preferido': 8.1, 'Escolhido': 7.5},
'detergente': {'Qbarato': 3.2, 'UltraK': 4.2, 'Market': 3.8, 'Preferido': 3.3, 'Escolhido': 3.9},
'cerveja': {'Qbarato': 6.4, 'UltraK': 6.4, 'Market': 6.5, 'Preferido': 6.5, 'Escolhido': 6.4},
'manteiga': {'Qbarato': 8.7, 'UltraK': 8.7, 'Market': 9.2, 'Preferido': 8.9, 'Escolhido': 8.6}}
```

14. O dia e horário de matrícula dos calouros da PUC depende do curso desejado. Faça uma função que receba dois dicionários:

- dicionário de cursos: para cada nome do curso há associado uma lista com a data de matrícula e o número de aprovados no curso.
- dicionário de aprovados: para cada nome de aluno, há uma lista com o nome do curso e o sua colocação no vestibular

Esta função deve exibir para cada aluno, seu nome, seu curso, dia e horário de matrícula. Os alunos são divididos em 4 grupos de acordo com sua colocação no curso. O 1º grupo é composto pelos melhores colocados (1/4 do total de vagas), o 2º grupo pelos seguintes melhores colocados (1/4 seguinte do total de vagas) e assim sucessivamente.

Por exemplo: um curso com 100 vagas, do 1º ao 25º colocados pertencem ao grupo 1, do 26º ao 50º colocados, pertencem ao grupo 2, o 51º ao 75º colocados pertencem ao grupo 3 e do 76º ao 100º colocados pertencem ao grupo 4.

Há 4 horários possíveis:

- ✓ 1º horário - 8h às 11:30h:
- ✓ 2º horário: 12h às 13:30h
- ✓ 3º horário: 14h às 15:30h
- ✓ 4º horário: 16h às 17:30h

15. Batalha Naval simples:

a. Monte um dicionário com os tipos de embarcações:

- 'S': submarinos, 3 posições, pontos 30
- 'D': destroyers, 5 posições, pontos 50
- 'C': cruzadores, 7 posições, pontos 100

b. Monte um dicionário com a localização das embarcações.

- tupla da posição no mapa da embarcação: sigla da embarcação

Observação: Há apenas uma embarcação de cada tipo

Jogo:

- O usuário digita as coordenadas separadas por virgula. (-1,-1) para finalizar
- Caso tenha acertado alguma embarcação, o programa exibe o nome da embarcação, caso contrário exibe 'Água'.
- Quando uma embarcação é totalmente destruída, o programa mostra os pontos obtidos.

Término do programa:

- quando o usuário desiste: coordenadas -1,-1
- todas as embarcações foram destruídas.

Em ambos os casos, deve ser exibido o total de pontos