**Lista de Exercícios: Listas são dinâmicas**

1. Considere as seguintes listas:

L1= [1,2,2,2,2,2,2,4,4,5,5,6,7,8,8,8,9,10,10]

L2=[['A', [[1, 2, 3], [4, 5, 6], [[7, 8], 9]]], ['B', [[3, 2, 3], [5, 5, 6], [[7, 7], 9]]], ['B', [[1, 2, 3], [4, 5, 6], [[7, 8], 9]]],

['C', [[1, 2, 3], [4, 5, 6], [[7, 8], 9]]], ['A', [[1, 2, 3], [4, 5, 6], [[7, 8], 9]]]]

L3= ['Ana Maria', 'Bola Bon', 'Pedro Joaquim Jo', 'Maria', 'Zé Zangado', 'Carlos Sá', 'Carmen Carla Carim Costa']

**Com a L1:**

1. Função A: recebe a lista L1, construa uma cópia em LAUX e retire os elementos repetidos de LAUX. Lembre-se (e teste se quiser) que se usar o comando de atribuição, a L1 também será modificada. Utilize a cópia por slice ou método .copy()

**Com a L2:**

1. Função B: recebe a L2 e retorna uma lista com as letras de L2 SEM REPETIÇÃO.
2. Função C1: recebe a L2 e retorna uma lista onde cada elemento é [Letra, Total]. Total é a soma dos números em cada elemento da L2. (Tem repetição)
3. Função C2: recebe a lista gerada pela função C1 e retorna uma lista onde cada elemento é [Letra, TotalGeral]. Total é a soma de todos os números associados à cada letra. (Sem repetição de Letra)

**Com a L3:**

1. Função D: recebe a L3 e exibe quantos caracteres há em cada nome
2. Função E: recebe a L3 e exibe quantas palavras há em cada nome. Dica: use o método **.split** de strings (dado uma string e um caractere separador, gera uma lista cujos elementos são as partes *cortadas* da string.
3. Função F: recebe a L3 e exibe o nome com maior número de palavras da L3. Dica: use o método .split de strings
4. Função G: recebe a L3 e exibe os elementos da lista (nomes completos) em que todas as partes se iniciam pela mesma letra. Dica: use o método .split de strings e acesse o primeiro caractere dos elementos da lista resultante
5. Função H: recebe a L3 e um nome e exibe a posição deste nome em L3 ou a mensagem “Não Existe”. Dica: tente usar o método .index de lista
6. Algumas escolas participaram de uma competição em diversas modalidades esportivas. O resultado deve ser armazenado em uma lista onde cada elemento possui a seguinte estrutura:

[ aluno, escola, [tipo da medalha,...,tipo da medalha]]

* aluno: inteiro;
* escola: string;
* O tipo da medalha: 1 caractere (‘o’ - ouro, ‘p’ – prata, ‘b’ – bronze).

1. Construa uma função que receba esta lista e retorne uma nova lista onde cada elemento possui as seguintes informações: [nome da escola, pontos totais]

Para testar esta função utilize a seguinte lista:

lRes=[[1010,'Esc1',['o','o','o','p','p']], [2010,'Esc2',['o','p','b','p','b']], [1020,'Esc1',['b','p']], [1030,'Esc1',['b']],

[3010,'Esc3',['o','o','o','o','p','o','o']], [2010,'Esc2',['b','b','b']], [3020,'Esc3',['o','o']]]

1. Construa um programa principal, que monte a lista de medalhas a partir de dados digitados pelo usuário e mostre:

* Para cada escola que participou da competição: o total geral de pontos da escola;
* O nome da escola campeã com seu total de pontos (considere que não há empate. DESAFIO: considere que há empate);

O total geral de pontos é calculado da seguinte forma:

* 1 medalha de ouro contabiliza 5 pontos;
* 1 medalha de prata contabiliza 3 pontos;
* 1 medalha de bronze contabiliza 1 ponto.

Para testar a função do item a) utilize a seguinte lista:

lRes=[[1010,'Esc1',['o','o','o','p','p']],[2010,'Esc2',['o','p','b','p','b']],[1020,'Esc1',['b','p']], [1030,'Esc1',['b']],

[3010,'Esc3',['o','o','o','o','p','o','o']], [2010,'Esc2',['b','b','b']], [3020,'Esc3',['o','o']]]

1. Um restaurante colocou um computador na entrada para controlar a ocupação das suas mesas. Quando um cliente chega e não tem mesa vazia, seu nome fica armazenado numa lista. Quando uma mesa é liberada, um nome (pense qual...) é retirado da lista e o cliente entra.

Escreva a função controle( ) que recebe uma lista com o número das mesas disponíveis . Lê e trata as opções:

**C** – chegada de um cliente – Perguntar o nome e caso não tenha mesa, deve incluí-lo na lista e avisá-lo para aguardar. Caso haja uma mesa vazia, emitir a mensagem "Sr.(a) N , favor encaminhar-se a mesa m."; retirando a mesa da lista de mesas disponíveis;

**L** – libera uma mesa - lê um número de mesa (m) e caso tenha alguém esperando, emitir a mensagem "Sr.(a) N , favor encaminhar-se a mesa m."; caso contrário, incluir o número da mesa na lista de mesas disponíveis;

**F** – fim;

1. A administração de uma empresa de cooperativados deseja algumas referências cruzadas sobre seus prestadores de serviços terceirizados. Ela mantém as duas listas descritas a seguir:

**lTerc:** que contém os serviços prestados por terceiros, onde cada elemento contém as seguintes informações:

matrícula na cooperativa: inteiro

projeto alocado: nome do projeto

número de horas trabalhadas no projeto: inteiro

  ( há um elemento na lista para cada projeto que o prestador trabalha, logo pode ter repetição da matrícula do cooperativado nesta lista)

**lProj** contendo as seguintes informações:

nome do projeto: string

valor da hora trabalhada no projeto: real

situação: 1 (em andamento) ou 0 (finalizado)

A função *ConstroiLTerc*, retorna a lista de terceirizados e a função *ConstroiLProj*, retorna a lista de projetos.

Após a construção destas listas, um menu deve ser apresentado continuamente na tela, com as seguintes opções (devem ser explicadas brevemente na tela):

**Opção 1:**  chamar uma função que liste os prestadores de serviço: matrícula na cooperativa, salário

**Opção 2:**  chamar uma função que lê do teclado uma matrícula na empresa do prestador de serviço e informe na tela: projetos alocados atualmente (nome, horas, montante)

**Opção 3:** chamar uma função que lê do teclado uma matrícula na cooperativa de prestador de serviço, um nome de projeto e uma quantidade de hora gastas no projeto e, caso o prestador de serviço já esteja vinculado ao projeto, atualize seu número de horas, senão, inclua este novo projeto na lTerc

**Opção 4:**  fim

def ConstroiLTerc():

lT= [[10,'P1',12],[10,'P2',12],[20,'P1',10],[30,'P1',10],

[40,'P3',10],[10,'P3',10],[20,'P2',12],[30,'P3',12]]

return lT

def ConstroiLProj():

lP= [['P1',100, 1],['P2',200,1],['P3',100,0],['P4',500,0]]

return lP

1. Um jogo individual se baseia nos resultados obtidos em n lançamentos de um par de dados virtuais, de tal forma que o jogador ganha k pontos cada vez que obtém uma seqüência de k números seguida de sua seqüência reversa. Por exemplo, se em 15 lançamentos o jogador obtiver os números {4, 3, 9, 11, 5, 7, 2, 9, 9, 2, 7, 10, 6, 6, 8} ele vai obter 4 pontos, pois entre os números obtidos está uma seqüência de 3 números (7, 2 e 9) seguida de sua seqüência reversa (9, 2 e 7) e uma seqüência e sua reversa de apenas 1 número (6).

Crie uma função para computar quantos pontos um jogador obteve a partir de uma lista recebida que armazena a seqüência de números obtidos. O retorno da função deve ser o total de pontos obtido pelo jogador.

Dica: Pense em 'empilhar' os valores em uma nova lista: antes de incluir um número nesta lista, verifique se ele é igual ao último incluído. Se for, retire o último e conte um ponto

1. Considere um programa de gerenciamento comercial que armazene as informações referentes aos dados sobre os produtos do estoque de uma loja em uma lista onde cada elemento possui: o código do produto e uma lista com a data de validade de cada uma das unidades em estoque deste produto.

a) Escreva uma função que receba como parâmetros a lista de estoque de produtos ( lValUnidadesEmEstoque) e uma data. Esta função deve retirar desta lista as unidades dos produtos que estão fora da validade, isto é, cuja data de validade seja anterior à data recebida. Caso algum produto fique com estoque zerado, isto é, todas as suas unidades estão fora da validade, deve ser retirado desta lista e exibido.

b) Construa um programa para testar sua função

Exemplo:

Para a lista inicial de produtos:

lValUnidadesEmEstoque = [

[10,['12/12/2017', '12/12/2017','12/1/2017', '12/3/2017', '12/6/2017']],

[20,['12/1/2017', '12/2/2017','12/3/2017', '2/7/2017', '22/5/2017','12/8/2017']],

[30,['12/1/2017', '12/1/2017','12/1/2017']],

[40,['12/12/2017', '12/12/2017','12/12/2017', '12/12/2017', '12/12/2017']]

]

e data '18/6/2017'

A lista de produtos final é:

lValUnidadesEmEstoque = [ [10,['12/12/2017', '12/12/2017']],

[20,['2/7/2017','12/8/2017']],

[40,['12/12/2017', '12/12/2017','12/12/2017', '12/12/2017', '12/12/2017']]

]

1. *Outra versão da ex. 3.* Um restaurante colocou um computador na entrada para controlar a ocupação das suas mesas. As mesas são identificadas por uma cor e o número de lugares do seguinte modo: “cnn”, onde c é uma letra indicando a cor da área da mesa no salão e nn é a quantidade de lugares. Quando um cliente chega e não tem mesa vazia para a quantidade de lugares desejada, seu nome fica armazenado numa lista, concatenado à quantidade de lugares desejados. Quando uma mesa é liberada, um nome que deseja a quantidade de lugares é retirado da lista e o cliente entra.

Escreva a função controle( ) que recebe uma lista com o número das mesas disponíveis . Lê e trata as opções:

**C** – chegada de um cliente – Perguntar o nome e quantidade de lugares e caso não tenha mesa, deve incluí-lo na lista e avisá-lo para aguardar. Caso haja uma mesa vazia, emitir a mensagem "Sr.(a) *N* , favor encaminhar-se a mesa *cnn*."; retirando a mesa da lista de mesas disponíveis;

**L** – libera uma mesa - lê um número de mesa (cnn) e caso tenha alguém esperando, emitir a mensagem "Sr.(a) *N* , favor encaminhar-se a mesa *cnn*."; caso contrário, incluir o número da mesa na lista de mesas disponíveis;

**F** – fim;