

Lista 4 - Listas

27 de novembro de 2018

Disciplina: Linguagem de Programação I
Prof. Tiago Eugenio de Melo
Período: 2018/2

Os exercícios têm uma classificação de acordo com o nível de dificuldade. O objetivo dessa classificação é orientar os alunos nas escolhas por quais exercícios eles devem começar a resolver. Quanto maior o número de estrelas mais difícil será o exercício. A legenda seguirá o padrão abaixo:

★ Esses são os exercícios mais simples e que devem ser realizados no início do estudo de cada assunto. Esses são os exercícios triviais.

★★ Esses são os exercícios modificados da classificação anterior com o acréscimo de um certo grau de dificuldade. Dessa forma, esses exercícios deixam de ser triviais.

★★★ Esses são os exercícios mais complexos que exigem uma leitura mais minuciosa do exercício. Normalmente esses exercícios possuem mais de uma solução. Também estão nesta categoria os exercícios que não são tão complexos (em relação ao grau de dificuldade), mas que devido à sua extensão, acabam por se tornar mais difíceis que os exercícios das categorias anteriores.

★★★★ Esses exercícios são os que estão na categoria com o maior grau de dificuldade. Normalmente, envolvem algum desafio de lógica de programação ou então pelo tamanho final do algoritmo (ou programa). Os exercícios devem ser resolvidos em Python 2.7.

1. ★ Faça um programa que leia um vetor de 5 números inteiros e mostre-os.
2. ★ Faça um programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.
3. ★ Faça um programa que leia 4 notas, mostre as notas e a média na tela.
4. ★ Faça um programa que leia um vetor de 10 caracteres, e diga quantas consoantes foram lidas. Imprima as consoantes.

5. ★ Faça um programa que leia 20 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números IMPARES no vetor ímpar. Imprima os três vetores.
6. ★ Faça um programa que leia um vetor de 5 números inteiros, mostre a soma, a multiplicação e os números.
7. ★ Faça um programa que peça a idade e a altura de 5 pessoas, armazene cada informação no seu respectivo vetor. Imprima a idade e a altura na ordem inversa a ordem lida.
8. ★ Faça um programa que receba uma lista e exiba os elementos da última metade na frente dos elementos da primeira metade.
9. ★ Faça um programa que leia um vetor A com 10 números inteiros, calcule e mostre a soma dos quadrados dos elementos do vetor.
10. ★★ Faça um programa que receba uma lista de números armazenados de forma crescente, e dois valores (limite inferior e limite superior), e exiba a sublista cujos elementos são maiores ou iguais ao limite inferior e menores ou iguais ao limite superior.
Exemplo: lista inicial=[12,14,15,16,18,20,24,26,28,32,34,38]
limite inferior=13
limite superior = 26
lista exibida: [14,15,16,18,20,24,26]
11. ★ Faça um programa que leia dois vetores com 10 elementos cada. Gere um terceiro vetor de 20 elementos, cujos valores deverão ser compostos pelos elementos intercalados dos dois outros vetores.
12. ★ Altere o programa anterior, intercalando 3 vetores de 10 elementos cada.
13. ★ Foram anotadas as idades e alturas de 30 alunos. Faça um programa que determine quantos alunos com mais de 13 anos possuem altura inferior à média de altura desses alunos.
14. ★ Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em uma lista. Após isto, calcule a média anual das temperaturas e mostre todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . .).
15. ★★ Faça uma função que:
 - receba duas listas e exiba a união destas listas.
 - receba duas listas e exiba a interseção destas listas.

- receba duas listas e exiba a intercalação destas listas, isto é, 1º da 1ª lista, 1º da 2ª lista, 2º da 1ª lista, 2º da 2ª lista...
16. ★★ Faça um programa que receba uma lista qualquer e retorne True se ela está ordenada ou False, caso contrário. Lembre-se que não é possível comparar elementos de tipos distintos. Caso o elemento seja uma sublista, a sublista deve ser verificada.
17. ★★★ Crie um programa que recebe uma lista de strings e:
- retorne o elemento com mais caracteres.
 - retorne a média de vogais nos elementos (número de vogais de cada elemento/número de elementos).
 - retorne o número de ocorrências do primeiro elemento da lista.
 - retorne a palavra lexicograficamente maior.
 - conte o número de ocorrências de palavras compostas.
 - retorne a quantidade de vizinhos iguais.
18. ★★ Faça um programa que percorre uma lista e exiba na tela o valor mais próximo da média dos valores da lista.
- Exemplo:
- lista = [2.5, 7.5, 10.0, 4.0] (média = 6.0)
- Valor mais próximo da média = 7.5
19. ★★★ Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são: "Telefonou para a vítima?" ; "Esteve no local do crime?"; "Mora perto da vítima?"; "Devia para a vítima?"; "Já trabalhou com a vítima?"
- O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".
20. ★★ Faça um programa que percorre uma lista com o seguinte formato: [['Brasil', 'Italia', [10, 9]], ['Brasil', 'Espanha', [5, 7]], ['Italia', 'Espanha', [7,8]]]. Essa lista indica o número de faltas que cada time fez em cada jogo. Na lista acima, no jogo entre Brasil e Itália, o Brasil fez 10 faltas e a Itália fez 9. O programa deve imprimir na tela: a) o total de faltas do campeonato b) o time que fez mais faltas c) o time que fez menos faltas