



Lista de exercícios 5 – Strings e Listas

1) Escreva uma função **inverte(s)** que recebe uma string **s** e retorna a os caracteres da string de trás para frente. Não use a função `reverse` da linguagem Python. Ex:

inverte("Russas") → "sassuR"

2) Escreva uma função **palindromo(s)** que retorna verdadeiro ou falso, caso a string entrada seja um palíndromo. (Palíndromos são palavras ou frases que são iguais tanto da esquerda pra direita como da direita pra esquerda).Ex:

palindromo("bemtevi") → False

palindromo("arara") → True

3) Escreva uma função **removePedaco(s,p)** que recebe uma string **s** e remove todas as vezes em que a string **p** aparece dentro de **s**. Ex:

removePedaco("Universidade Federal","idade") → "Univers Federal"

4) Escreva uma função **contaVogais(s)** que retorna a quantidade de vogais presente na string **s**. Ex:

contaVogais("Universidade") → 6

5) Escreva uma função **repeteVogal(s)** que retorna a string **s**, mas com as vogais repetidas. Ex:

repeteVogais("Universidade") -> "UUniveersiidaadee"

6) Seja a seguinte lista: **minha_lista = [76, 92.3, "oi", True, 4, 76]**. Após criar essa lista, faça o que se pede abaixo:

1. Insira a string “pêra” e o número 76 no final da lista
2. Insira a string “gato” na posição 3 da lista
3. Insira o número 99 no início da lista
4. Escreva o índice onde a string “oi” se encontra dentro da lista
5. Conte quantas vezes o número 76 aparece na lista
6. Remova a primeira ocorrência do número 76 da lista
7. Descubra onde o número 4 está na lista e o remova.

7) Escreva um programa que armazene 10 números inteiros lidos do usuário em uma lista e depois escreva os valores armazenados.

8) Escreva uma função **busca(l, x)** que procure o número **x** dentro da lista **l** e retorne um valor booleano indicando se **x** está dentro da lista de números ou não. Não é permitido utilizar o operador **in** nessa questão. Ex:

busca([2,4,6,8],4) → True

busca([2,4,6,8],3) → False

9) Crie uma função **numerosAleatorios(n, min, max)** que retorna uma lista com **n** números inteiros entre os valores **min** (incluso) e **max** (não inclusivo). Ex:

numerosAleatorios(5,10,30) → [13,29,23,13,21]

10) Escreva uma função **media(l)** que recebe uma lista **l** de números e retorna a média dos valores da lista. Não utilize funções prontas de somatório ou média dentro de sua função.

11) Escreva uma função **limites(l)** que recebe uma lista **l** de números e retorna uma tupla (**min,max**), onde **min** representa o menor dos elementos da lista e **max** o maior dos elementos da lista. Não utilize funções prontas para determinar o menor ou maior elemento da lista.

12) Escreva uma função **contaPares(l)** que recebe lista **l** de números inteiros e retorna a quantidade de números pares presente na lista.

13) Escreva uma função **filtraPares(l)** que recebe uma lista **l** de números inteiros e retorna uma lista contendo apenas os números pares da lista entrada.

14) Escreva uma função **slice(l,i,f)** que recebe uma lista **l** e dois números inteiros **i** e **f** e retorna a mesma coisa que **l[i:f]**, mas sem usar as slices da linguagem Python.

15) Escreva uma função **menorFrente(l)** que recebe uma lista de números e retorna a mesma lista com a seguinte alteração: o menor número da lista está posicionado na primeira posição. Ex: **menorFrente([10, 6, 7, 1, 2, 4, 11, 24])** → **[1, 10, 6, 7, 2, 4, 11, 24]**

16) Escreva uma função **ordenacaoSelecao(l)** que reutiliza a função da questão anterior para ordenar uma lista de números. Não é permitido utilizar funções como **sort**. Ex: **ordenacaoSelecao([10, 6, 7, 1, 2, 4, 11, 24])** → **[1, 2, 4, 6, 7, 10, 11, 24]**

17) Escreva uma função **divideNumeros(l,p)** que retorna a tupla (**menores,maiores**), onde **menores** é uma lista contendo todos os elementos da lista **l** menores do que **p** e **maiores** é uma lista contendo todos os elementos da lista **l** maiores ou iguais a **p**.

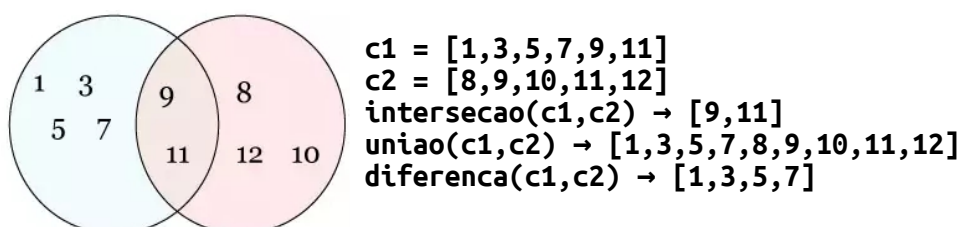
18) Escreva uma função **combina(l1,l2)** que recebe duas listas de números **l1** e **l2**. Assuma que ambas as listas já tenham seus valores ordenados. Agrupe os elementos de ambas as listas em uma única lista, contendo todos os valores em ordem crescente, e retorne essa lista.

19) Escreva uma função **removeRepetidas(l)** que recebe uma lista de valores e deixa apenas uma cópia de cada elemento, caso haja valores repetidos.

20) Escreva as seguintes funções que simulam as operações entre conjuntos:

- **intersecao(c1,c2)**
- **uniao(c1,c2)**
- **diferenca(c1,c2)**

Suponha que as listas usadas como entrada não possuem elementos repetidos. Garanta que após a operação, as funções retornarão conjuntos sem elementos repetidos. Ex:



21) Escreva uma função **citacao(nome)** que receba o nome de uma pessoa e retorne ela em formato de citação bibliográfica, ou seja, o último sobrenome aparece primeiro separado por vírgula pelas diversas iniciais dos outros nomes na ordem em que elas aparecem originalmente no nome. Dica: leia a documentação das funções `split` e `join` que são aplicadas sobre strings. Ex:
citacao("Rafael Fernandes Ivo") → "Ivo, R. F."