

## Exercícios sobre listas em Python

O problemas desta lista devem ser resolvidos com o uso de strings e duplas em Python. Caso você utilize funções, para cada um dos exercícios, crie um programa main para executar suas funções. **OBS.: utilize as mesmas regras das listas anteriores para dar nomes aos arquivos.**

### Questões:

1. **Ordem Crescente.** Escreva um programa Python que leia números inteiros (do usuário) e os armazena em uma lista. Seu programa deve continuar lendo inteiros até que o usuário entre com o valor 0 (zero). Então, o programa deve exibir em ordem **crescente** todos os números digitados pelo usuário sem incluir o valor 0, com um valor em cada linha impressa. Obs.: você pode implementar o algoritmo de ordenação BubbleSort ou usar o método **sort** ou a função **sorted** para ordenar a lista.
2. **Ordem decrescente.** Escreva um programa Python que leia números inteiros (do usuário) e os armazena em uma lista. Seu programa deve continuar lendo inteiros até que o usuário entre com o valor 0 (zero). Então, o programa deve exibir em ordem **decrescente** todos os números digitados pelo usuário sem incluir o valor 0, com um valor em cada linha impressa.
3. **Evitando duplicatas.** Crie um programa Python que leia palavras do teclado até que o usuário forneça uma palavra vazia (somente a tecla enter). Depois disso, seu programa deve mostrar somente uma vez cada palavra digitada pelo usuário. Ou seja, se o usuário forneceu mais de uma vez a mesma palavra, ela só poderá ser exibida uma vez. As palavras devem ser exibidas na mesma ordem em que foram digitadas. Por exemplo, se o usuário digitar:  
Primeira  
Segunda  
Primeira  
Terceira  
Segunda  
  
Então seu programa deve exibir:  
Primeira  
Segunda  
Terceira
4. **Negativos, zeros e positivos.** Escreva um programa Python que leia números inteiros até que uma linha em branco seja fornecida pelo usuário (se ele digitar somente enter). Depois que todos os inteiros tiverem sido lidos, seu programa deve mostrar todos os números negativos, seguidos por todos os zeros e depois todos os números positivos. Dentro de cada grupo os números devem ser exibidos na ordem em que foram fornecidos pelo usuário. Por exemplo, se o usuário forneceu os valores 3, -4, 1, 0, -1, 0 e -2, então seu programa deve exibir os valores -4, -1, -2, 0, 0, 3 e 1, cada um em uma linha.
5. **Lista de divisores.** Um divisor de um numero inteiro n é um número inteiro positivo menor que n, tal que divida n em partes inteiras (divisão exata, sem resto). Escreva uma função Python que calcula todos os divisores de um número inteiro positivo. A função deve retornar

uma lista contendo todos os divisores. Escreva uma função main para demonstrar o funcionamento da sua solução, a função main deve ler um número do usuário e imprimir todos os seus divisores.

6. **Números perfeitos.** Um número inteiro positivo  $n$  é chamado de número perfeito se a soma de todos os divisores de  $n$  é igual a  $n$ . Por exemplo, 28 é um número perfeito porque seus divisores são 1, 2, 4, 7 e 14; e  $1+2+4+7+14=28$ . Escreva uma função para verificar se um número é perfeito. A função deve receber somente um número inteiro positivo e retornar True se ele for perfeito ou False caso contrário. Escreva uma função main para identificar e imprimir todos os números perfeitos de 1 a 10.000. Obs.: você pode usar o código do exercício anterior para lhe ajudar nesta tarefa.
7. **Somente palavras.** Neste exercício você deve criar uma função em Python que recebe um texto em uma string e retorna uma lista somente com as palavras, sem espaços ou símbolos de pontuação. Por exemplo, se a string for: "Beleza! Este é um ótimo exemplo, você não acha?", sua função deveria retornar a seguinte lista: ["Beleza", "Este", "é", "um", "ótimo", "exemplo", "você", "não", "acha"]. Escreva uma função main que demonstre o funcionamento da sua solução.
8. **Formatando uma lista.** Quando escrevemos uma lista em português, geralmente separamos os itens por vírgula e colocamos a conjunção "e" entre os dois últimos itens, a não ser que a lista só tenha um item. Considere as listas abaixo:

```
maças  
maças e laranjas  
maças, laranjas e bananas  
maças, laranjas, bananas e limões
```

Escreva uma função que receba como parâmetro uma lista de strings e retorne uma única string contendo todos os itens da lista formatados conforme mostrado acima. Apesar dos exemplos acima terem no máximo 4 itens, sua função deve se comportar corretamente para listas de qualquer tamanho. Escreva uma função main demonstrando o funcionamento de sua função.

9. **Contagem de elementos.** A biblioteca padrão de funções do Python possui um método chamado count, que determina quantas vezes um determinado valor ocorre em uma lista. Neste exercício você deve criar uma nova função chamada countRange que deve determinar e retornar a quantidade de elementos em uma lista que são maiores ou iguais a um valor mínimo e menores que um valor máximo. Sua função deve receber três parâmetros: a lista (de números), o valor mínimo e o valor máximo. Ela deve retornar um valor inteiro maior ou igual a zero. Escreva uma função main demonstrando o funcionamento de sua função.