

## Mostrando Anagramas

### Mário Leite

...

Um dos assuntos mais importantes na programação é a manipulação de “pedaços” de textos; de chamados *strings*. A maioria das linguagens oferece recursos bem eficientes para tratar *strings*: procurar *substring* numa dada *string*, verificar se determinado caractere está contido na *string*, inverter uma *string*, calcular o tamanho de uma *string*, converter toda a *string* em letras maiúsculas ou em minúsculas, etc. Estes dois últimos recursos, moralmente, são conseguidos diretamente com funções e/ou métodos próprios da linguagem; mas tem casos que, dependendo da linguagem implementada, são necessárias várias linhas de código para se conseguir o resultado desejado. Por exemplo: “*listar todos os anagramas de uma determinada palavra*”; se for tentado implementar isto em C ou em Delphi o código fica muito complicado, necessitando de muitas linhas de código. Como exemplo, digamos que seja necessário mostrar todos os anagramas de **PATO** (que no caso são 24 - permutação simples de 4). O algoritmo para fazer isto pode ser baseado no fato desta palavra ser um *array*:

- P>A>T>O ==> A>P>T>O ==> A>T>P>O ) ==> anagrama

Até que a letra **P** se torne a última letra do *array*.

Quando isso acontecer o processo se repete com a primeira letra, que agora será A>T>O>P; e assim, até que a palavra **PATO** retome á sua forma original (um caso).

Embora seja um ótimo exercício para os programadores iniciantes, é muito trabalhoso e até entediante para um programador experiente que deseja obter anagramas apenas como um meio de alcançar um objetivo mais “nobre” dentro do programa. Já no Python esta tarefa é muito simples de fazer com métodos da biblioteca **permutations**. O programa “**MostraAnagramas1**” mostra como exibir todos os anagramas de uma palavra digitada pelo usuário, e a **figura 1** mostra a saída do programa usando a palavra **PATO** como exemplo. E se for preciso mostrar os anagramas com um determinado caractere fixado numa dada posição esta biblioteca oferece métodos que tornam a solução muito fácil

```
'''
MostraAnagramas1.py
-----
Mostra anagramas de uma palavra digitada.
-----
Autor: Mário Leite
Data: 16/04/2023
-----
'''
from itertools import permutations

print()
palavra = input("Digite a palavra: ")
palavra = palavra.upper() #converte a 'palavra' em maiúsculas

#Faz as permutações da palavra digitada
permutas = list(set([''.join(p) for p in permutations(palavra)]))

#Mostras os anagramas da palavra digitada
print()
print("Anagramas da palavra {}: ".format(palavra))
for anagrama in permutas:
    print(anagrama)
#Fim do programa "MostraAnagramas" -----
```



```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> == RESTART: G:/BackupHD/HD-D/Livros/Livroll/Códigos/Nível 1/MostraAnagramas.py =
Digite a palavra: pato
Anagramas da palavra PATO:
TPOA
TAOP
OATP
ATOP
AOPT
PTAO
OPAT
OAPT
APOT
OTPA
OPTA
AOTP
TOPA
POAT
PATO
OTAP
POTA
TPAO
TAPO
TOAP
APTO
PAOT
ATPO
PTOA
>>> |
```

Ln: 33 Col: 0

**Figura 1 - Um exemplo de saída do programa**