



Lista De Exercícios #04: PYTHON – String (Funções e Métodos)

Observações:

1. As respostas deverão ser submetidas no link correspondente a essa lista disponível no Moodle;
2. Os programas deverão ser desenvolvidos em linguagem PYTHON;
3. Cada questão deverá ser respondida em arquivos em separado.
4. Atentem para o prazo de submissão. Não serão aceitos envios posteriores a data limite

1. Faça um programa que conte a quantidade de vogais de uma *string*. Considerar também as vogais acentuadas.
2. Faça um programa que solicite ao usuário uma frase e substitua todos os espaços por (underline/sublinhado). O programa deverá exibir a frase original e a frase após as substituições.
3. Faça um programa que solicite ao usuário uma frase, uma palavra antiga e uma palavra nova. O programa deverá imprimir a frase digitada, as duas palavras digitadas (a antiga e a nova) e a frase substituindo a palavra antiga pela palavra nova.
4. Faça um programa que receba um valor qualquer e informe se esse valor digitado é um palíndromo (utilizar **laço de repetição WHILE**).

Lembrando que palíndromo é uma cadeia de caracteres que quando lida da esquerda para a direita é igual a quando se lê da direita para esquerda.

Exemplos: ARARA, 15051, ...

5. Faça um programa que receba duas palavras e informe se uma é anagrama da outra.
Lembrando que anagrama é uma palavra que é feita a partir da transposição das letras de outra palavra.
Exemplo: IRACEMA é um anagrama de AMERICA; ROMA é um anagrama de AMOR;...



6. Faça um programa que receba uma palavras e imprima essa palavra na vertical (em escada).

Exemplo: Ao ser informada a palavra **NATAL**, o programa imprime conforme poder ser visto abaixo:

N
NA
NAT
NATA
NATAL

7. Dado o código a seguir, implemente um programa que faça o mesmo utilizando **APENAS** um laço **WHILE**.

Código	Saída
<pre>1 variavel = '1234567890' 2 qt_caracteres = len(variavel) 3 4 posicao = 1 5 ~ while posicao <= qt_caracteres: 6 print(variavel[0:posicao]) 7 posicao += 1 8 9 posicao = qt_caracteres-1 10 ~ while posicao >= 0: 11 print(variavel[0:posicao]) 12 posicao -= 1</pre>	<pre>1 12 123 1234 12345 123456 1234567 12345678 123456789 1234567890 123456789 12345678 1234567 123456 12345 1234 123 12 1</pre>

8. Faça um programa que leia um CPF (considerar somente que sejam informados os 11 dígitos numéricos) e verificar se o CPF é válido (pesquisar fórmula de cálculo do dígito verificador do CPF).
9. Faça um programa que emule o jogo da forca. O programa terá uma constante chamada **PALAVRA_CHAVE** que armazenara a palavra a ser descoberta. O programa deverá solicitar ao usuário as letras e à medida que as letras forem sendo digitadas o programa irá exibir se o usuário acertou ou não. O jogo deverá considerar maiúsculas e minúsculas iguais. O jogador poderá errar 6 vezes antes de ser enforcado



10. Um robô pode se mover em oito sentidos em um plano cartesiano: **U** (cima); **D** (baixo); **R** (direita); **L** (esquerda); **O** (noroeste/cima-esquerda); **N** (nordeste/cima-direita); **E** (sudeste/baixo-direita) e **W** (sudoeste/baixo-esquerda).

Faça um programa que:

- (a) solicite ao usuário a posição inicial do robô (suas coordenadas X e Y);
- (b) solicite ao usuário uma *string*. Letras maiúsculas e minúsculas são indistintas e aquelas informadas que estejam fora das estabelecidas (**U, D, R, L, O, N, E e W**) devem ser ignoradas.
- (c) Com base em cada letra válida (**U, D, R, L, O, N, E e W**), o robô deverá se deslocar 1 (uma) unidade em cada eixo (X e Y) por vez em função da direção.

Ao final, indique:

- (a) a posição inicial do robô (coordenadas X e Y);
- (b) a posição final do robô (coordenadas X e Y);
- (c) quantos movimentos válidos ele executou;
- (d) quais foram os movimentos válidos que ele executou;
- (e) em que quadrante ele iniciou (posição inicial de X e Y) e;
- (f) em que quadrante ele terminou (posição final de X e Y)