Instruções:

a) As funções devem ser criadas em um arquivo separado (modularizadas);

1. Crie um programa que **peça ao usuário para digitar uma frase** e depois realize as seguintes operações a partir desta frase (todas usando funções que você deverá implementar, como determinado a seguir):
    - **Retorna as menores palavras**
        - Executa a função `menorPalavra`, informando na tela a menor palavra da frase
        - A função `menorPalavra` recebe como parâmetros a frase lida anteriormente e **retorna** uma lista contendo as menores palavras da frase (caso haja mais do que uma palavra com o tamanho – em quantidade de caracteres - da menor).

    - **Corrige M e N antes de P e B**
        - É sabido que antes de P e B não se usa N e sim M. Esta opção vai passar à função `corrigeMN` a frase lida anteriormente e vai retornar uma string contendo a frase corrigida.
        - A frase original e a frase corrigida deverão ser impressas (fora da função).
            - Obs: Considere que só haverá *np* ou *nb* como parte de uma palavra e não ocorrerão livremente.

2. **Apresenta quantidade de ocorrências das palavras na frase**
    - Solicite ao usuário uma lista contendo nomes aleatórios de objetos, separados por vírgulas, sendo que cada nome pode aparecer mais do que uma vez, em maiúsculas e/ou minúsculas, faça um programa que crie uma lista contendo o nome do objeto e quantas vezes ele aparece na lista, sem que haja repetição no nome dos objetos, que devem estar em minúsculo. No final da lista, deve-se guardar a quantidade total de produtos (soma das quantidades). Ao final, imprima um relatório de acordo com o modelo apresentado no exemplo.

      Exemplo de entrada do usuário: abajur, MESA, cadeira, Quadro, Abajur, caneta, PRATO, Prato, caneta, abajur, borracha, prato, caneta, Mesa, quadro, QUADRO, abajur, Caneta

      listaSaida=[['abajur', 4], ['mesa', 2], ['cadeira', 1], ['quadro', 3], ['caneta', 4], ['prato', 3], ['borracha', 1],18]

      Relatório que deve ser impresso:

      Quantidades de objetos cadastrados na lista:
      abajur - 4
      mesa - 2
      cadeira - 1
      quadro - 3
      caneta - 4
      prato - 3
      borracha - 1
      Total de objetos: 18

3. **Verifica se uma palavra é palíndromo.**
    - Solicita ao usuário que digite uma palavra e imprime na tela se a palavra é palíndromo ou não. Para isso executa o método `verificaPalindromo`.
        - Obs: Palíndromo é uma palavra que ao ser lida de traz pra frente é igual à palavra original. Exemplos de palíndromos: RADAR, OSSO, OVO, ASA.
4. Faça um programa que leia uma frase e apresente o percentual de vogais que existe nela.

5. Faça um programa simples de criptografia para codificar e decodificar mensagens digitadas pelo usuário, em que para criptografar, são feitas três passadas na mensagem:

1. na primeira passada os caracteres devem ser deslocados 3 posições para a direita, considerando caracteres da tabela ASCII. Por exemplo: letra 'a' deve virar letra 'd', letra 'E' deve virar letra 'H', letra 'z' deve virar caractere '}' e assim sucessivamente;
2. na segunda passada, a linha deverá ser invertida. Por exemplo, 'Av3Dc' deve virar 'cD3vA'
3. na terceira e última passada, o último caractere deve ser trocado com o primeiro.

Por exemplo, se a entrada for “abc123z”, o primeiro processamento sobre esta entrada deverá produzir “def456}”. O resultado do segundo processamento inverte os caracteres e produz “}654fed”. Por último, o quarto processamento produz o resultado final deve ser “d654fe}”.

O processo de decodificação é o inverso.

Devem ser implementadas duas funções:

1. recebe a mensagem em texto plano (não codificado) e retorna a mensagem criptografada (codificada)
2. recebe a mensagem criptografada (codificada) e retorna em texto plano (decodificada)

DICA – pesquise como se deslocar na tabela ASCII em Python