1. On suppose que l'on dipose d'une liste `dictionnaire` contenant tous les mots du dictionnaire. (Pour vos testes, créer une liste `dictionnaire` contenant trois mots : 'coucou', 'olivier', 'matrice')

   Ecrire une fonction `mots_de7lettres` qui retourne une liste ne contenant que les mots de 7 lettres du `dictionnaire`.

2. Ecrire une fonction `choix_mot`, qui choisit un mot de 7 lettres aléatoirement. (On pourra utiliser la fonction `len` qui prend en argument une chaine de caractères et qui retourne sa taille, (comme pour les listes))

3. Ecrire une fonction `transform` qui prend en argument une chaine de caractères `S` et retourne une liste dont chaque entrée est une lettre de la chaine `S`.

4. Créer une fonction `test_lettre` qui prend en argument un mot `M` et une lettre `a` et retourne la (ou les) position de la lettre `a` dans le mot `M`. Si `a` n'est pas dans le mot, la fonction retournera la liste vide.

   Exemples : `test_lettre`('olivier', 'i') –> [2, 4] et `test_lettre`('olivier', 'w') –> []

5. Ecrire une fonction `reponse` qui prend en argument deux chaines de caractères. L'une `M` correspondant au mot que l'on doit trouver et l'autre `P` correspondant à la proposition du joueur. Si le joueur propose une seule lettre alors la fonction `reponse` retourne la (ou les places) de la lettre dans le mot `M`, si le joueur propose un mot (donc plusieurs lettres) alors la fonction `reponse` retourne `True` si le mot est bon et une liste vide si le mot est mauvais. (La liste vide permet d'être dans le même cas que si on avait donné juste une lettre qui n'est pas dans le mot)

6. Ecrire une fonction `lettre_connue` qui prend en argument un mot `M` (correspondant au mot à trouver) et une liste `L` (qui correspond au position des lettres déjà trouvées) et qui retourne une chaine de caractères où les lettres dont la position sont dans `L` s'affiche en claire et sinon sont remplacées par des '*'

   Exemple : `lettre_connue` ('olivier', [1,2,5]) retourne '*li**i**' `lettre_connue` ('matrice', [0,1,2]) affiche 'ma*****'

7. Compléter le code suivant qui permet de jouer au jeu du pendu sans limite d'essais.

```
def pendu():
    mot_a_trouver=.....
    mot_propose=''
    list_lettres_connues=[]
    while mot_propose != ......:
        l=lettres_connues(.... , ....)
        # On affiche les lettres deja trouvees par le joueur)
        print(l)

        mot_propose = input('Donner une lettre ou une proposition
        #on demande au joueur une nouvelle lettre ou une nouvelle

        rep=reponse(...., .... )
        #on analyse la reponse.

        if ...==.... : #si le joueur a trouve le bon mot
            print( .... )# on le felicite
            return #on arrete le programme

        else: #sinon
            if len(rep)......:  #soit le mot n'est pas le bon ou
                print('essaye encore') #et on lui dit de reesayer

            else:
                print('il y a ' +..... +' lettre ' + .... )
                #on affiche le nombre de fois ou la lettre propos
                #apparait dans le mot cherche

                list_lettres_connues=....+....
                #et on ajoute a la liste des lettres
                #connues les nouvelles lettres.
```

8. Améliorer la fonction `pendu` pour que le programme s'arrête après 10 essais infructueux et affiche à chaque essais le nombre de tentatives restantes.