But

Le but est d'écrire un code `python` permettant de jouer au jeu du pendu.

Le jeu

Un joueur B (ou l'ordinateur) choisit un mot existant. Un joueur A essaye alors de deviner ce mot ; à chaque essai, le joueur A propose une lettre :

- ◇ si la lettre compose le mot à deviner, le joueur B dévoile les lettres correspondantes.
- ◇ si la lettre n'est pas dans le mot à deviner, le joueur B compte une erreur supplémentaire.

Si le joueur A a deviné toutes les lettres du mot à deviner en faisant moins de 6 erreurs, il gagne. Sinon à la 7è erreur, le joueur A a perdu et il est pendu :-(.

Aide

- ◇ Télécharger, dans votre répertoire élève, les fichiers `NSI-projet02-lf.txt` et `NSI-projet02.py` sur la page [http://forhan.maths.free.fr].
  Le fichier `NSI-projet02-lf.txt` contient plus de 22 000 mots de la langue français.
  Le fichier `NSI-projet02.py` définit la procédure `mot_alea() -> str` qui renvoie au hasard un mot du fichier `NSI-projet02-lf.txt` en majuscule et sans accent.
- ◇ Le code `python` ci-contre permet d'afficher un texte sans aller à la ligne

```
print("mon texte", end='')
```

- ◇ Le code `python` ci-contre permet de demander et renvoyer une chaîne de caractères en majuscules

```
lettre = input("sans accent").upper()
```

Étapes

a) Choisir un binôme et jouez deux fois au jeu du pendu (en échangeant les rôles).

b) Adapter le nom du fichier `NSI-projet02.py` au format `NSI-projet02-Nom1Nom2.py` et modifier la variable `PATH`.

c) Écrire une procédure `main() -> None`
qui comportera tout le code du jeu et qui définit le mot à déviner dans la variable `motADevine`.

d) Dans la fonction `main()`, définir un tableau `motDevine` qui contient autant de caractères '_' que le nombre de lettres du mot `motADevine` ainsi qu'un tableau `lettresErronees` initialement vide.

e) Écrire une procédure `afficheMotDevine(mDevin : list) -> None`
qui, si `mDevin=['i', '_', '_', 'o']`, écrira "mot deviné : i__o".
Tester votre fonction (penser à garder ces tests pour le contre-rendu) et appeler votre professeur.

f) Proposer une fonction `dejaProposee(lettre : str, mDevin : list, lError : list) -> bool`
qui renvoie `True` si le charactère `lettre` a déjà été proposée par le joueur A, `False` sinon.
Tester votre fonction (penser à garder ces tests pour le contre-rendu)

g) Proposer une fonction `nouvelleLettre(mDevin : list, lError : list) -> str`
qui affiche le mot `mDevin` en cours ainsi que le tableau des lettres erronées, puis demande à l'utilisateur une nouvelle lettre tant que celle-ci a déjà été proposée. La fonction renvoie au final la lettre en majuscule (on suppose que l'utilisateur n'entrera pas de lettre accentuée).
Tester votre fonction.

h) Proposer, puis tester, une fonction `indexLettre(lettre : str, mADevin : list) -> list`
qui renvoie un tableau contenant tous les indices où se trouve le caractère `lettre` dans le mot `mADevin`.

i) Proposer, puis tester, une fonction `notEgal(mADevin : str, mDevin : list) -> bool`
qui renvoie `True` si la liste `mDevin` possède au moins un caractère distinct de ceux de la chaîne de caractères `mADevin`, `False` sinon.

j) Proposer l'organigramme général qui met en œuvre chacune de ces fonctions pour définir votre fonction `main()`. Appeler votre professeur.

k) Terminer de coder la fonction `main()` et proposer un jeu de tests.

l) Approfondissement :
- ◇ modifier le code afin d'afficher le nombre d'essais à la fin du jeu
- ◇ modifier le code afin qu'un autre joueur puisse choisir le mot à deviner
- ◇ proposer une procédure `affichePotence(nbError : int) -> None` qui dessine une potence selon le nombre de lettres erronées.

expl1 : `affichePotence(2)` affichera

```
|
|
|
|_____
```

expl2 : `affichePotence(6)` affichera

```
|/   _0_
|     |
|
|_____
```

Rendu

À la fin du temps imparti, chaque groupe rendra 2 travaux :

a) un compte-rendu (au maximum 2 pages) contenant l'organigramme du code général, un résumé de votre projet, les jeux de tests effectués ainsi que leurs résultats (ceux-ci pourront être mis dans votre fichier .py)

b) le fichier contenant le code `python` sera envoyé sous format numérique via l'ENT