

MANIPULATION FICHER TEXTE

MP/MP* LYCÉE CARNOT

1. RÉVISION DE COURS : INTRODUCTION DES NOTIONS AVEC LA COMPRÉHENSION D'UN PROGRAMME DE JEU DU PENDU

1.1. **Règles du jeu.** Le jeu du pendu consiste à retrouver un mot le plus vite possible (avant que le dessin du pendu soit terminé) en proposant des lettres. Si la lettre appartient au mot, elle est écrite aux bons emplacements, sinon on continue de dessiner le pendu. Nous allons étudier une version (non graphique) où il s'agira de deviner un mot avec le moins d'essais possible.

1.2. Programme du jeu du pendu. Exemple de programme en Python

```
# Le jeu du pendu

from random import choice

fichier = open("liste_mots.txt", "r")
liste_mots = fichier.readlines()
mot = choice(liste_mots)
mot = mot.rstrip()
fichier.close()

mot_devine = "-" * len(mot)
print(mot_devine)
nbr_essais = 0

while mot_devine != mot:
    lettre = input("Entrez une lettre ou '?' pour abandonner : ")
    lettre = lettre[0]
    if lettre == '?':
        print('Le mot était',mot)
        break
    lettre = lettre.upper()
    for i in range(len(mot)):
        if lettre == mot[i]:
            mot_devine = mot_devine[:i] + lettre + mot_devine[i+1:]
    print(mot_devine)
    nbr_essais += 1

if mot == mot_devine:
    print('Bravo ! Le mot',mot,'a été trouvé en',nbr_essais, 'coups')
```

1.3. Analyse détaillée du programme.

Exercice 1. Reprendre les parties du programme suivantes et analyser les, comme pour la partie suivante :

```
from random import choice
```

La fonction `choice` permet de choisir un élément au hasard dans une liste. Ici, il s'agira de choisir un mot.

```
fichier = open("liste_mots.txt", "r")
liste_mots = fichier.readlines() # ...
mot = choice(liste_mots) # ...
mot = mot.rstrip() # supprime le caractère "saut à la ligne"
fichier.close()
```

Solution.

```
mot_devine = "-" * len(mot)
print(mot_devine)
nbr_essais = 0
```

Solution.

```
while mot_devine != mot:
```

Solution.

```
lettre = input("Entrez une lettre ou '?' pour abandonner : ")
lettre = lettre[0] # évite des erreurs si un mot
# est entré au lieu d'une lettre
```

```
if lettre == '?':
    print('Le mot était',mot)
    break
```

Solution.

```
lettre = lettre.upper()
```

Solution.

```
for i in range(len(mot)):
    if lettre == mot[i]:
        mot_devine = mot_devine[:i] + lettre + mot_devine[i+1:]
    print(mot_devine)
    nbr_essais += 1
```

Solution.

```
if mot == mot_devine:
    print('Bravo ! Le mot',mot,'a été trouvé en',nbr_essais, 'coups')
```

Solution.

2. TD/TP : ADAPTATION À D'AUTRES JEUX DE LETTRES

Exercice 2 (Anagrammes). On veut réutiliser le fichier `liste_mots.txt` pour faire un jeu qui consiste à retrouver un mot, l'ordinateur ayant écrit ses lettres par ordre alphabétique.

Compléter le programme ci dessous pour obtenir ce jeu. (Ou écrire le programme ?)

```
# Anagrammes

from random import choice

fichier = open(.....)
liste_mots = ...           # met tous les mots du fichiers dans une liste
mot = ....                # prend au hasard un mot dans la liste
mot = mot.rstrip()        # supprime le caractère "saut à la ligne"
....

liste_lettres = .....
liste_lettres.sort()
anagramme = ""
for lettre in liste_lettres:
    anagramme = .....
print(anagramme)

mot_devine = input("Entrez l'anagramme : ")
mot_devine = mot_devine.upper()
if .....
    print('Non, le mot était',mot)
else:
    print('Bravo !')
```

Pour les questions suivantes, on pourra aussi utiliser des ensembles. Les ensembles en Python sont les mêmes ensembles qu'en mathématiques. On peut leur appliquer les mêmes opérations : union, intersection, différence symétrique, différence. On peut aussi tester si un ensemble est inclus dans un autre.

Le programme suivant résume les différentes opérations sur les ensembles.

```
ens1 = set([1, 2, 3, 4, 5, 6])
ens2 = set([5, 6, 7, 8])
ens3 = set([2, 4, 6])
# union
print(ens1 | ens2)
# intersection
print(ens1 & ens2)
# différence symétrique
print(ens1 ^ ens2)
# différence
print(ens1 - ens2)
# inclusion
```

```

print(ens2.issubset(ens1))
print(ens3.issubset(ens1))
# transformation en une liste
liste1 = list(ens1)
print(liste1)

set('carnaval')

```

Exercice 3. Écrivez une fonction qui remplace les voyelles d'un texte (éventuellement accentuées) par une étoile. Par exemple "Il était une fois" deviendra "*l *t**t *n* f**s".

Exercice 4. Pour les questions suivantes, on utilisera le fichier `dico.txt`.

- (1) Calculez le pourcentage de mots français où la seule voyelle est le "e" (il peut y en avoir plusieurs dans le mot). Par exemple : exemple, égrener ..
- (2) Affichez les mots de moins de 10 lettres ayant comme lettre centrale un "w". Le mot doit avoir un nombre impair de lettres. Par exemple : edelweiss, kiwis.
- (3) Affichez les mots de moins de 5 lettres qui commencent et finissent par la même lettre. Par exemple : aura, croc, dard.
- (4) Affichez tous les mots palindromes. Par exemple : serres, kayak, ressasser.

Exercice 5. Pour toutes les questions ci-dessous, utilisez le fichier `dico.txt`. Dans le jeu du pendu, les mots les plus difficiles à trouver sont ceux qui ont peu de voyelles et des consonnes rares. Voici les lettres de la plus fréquente à la moins fréquente, en français :

Lettres courantes : E A I S T N R U L O D M P
Lettres rares : C V Q G B F J H Z X Y K W

- (1) Affichez la liste des mots n'ayant aucune des lettres E A I S T N R U. Par exemple : transports, check-list.
- (2) Affichez la liste des mots commençant par au moins 4 consonnes consécutives (tiret non autorisé).