Université De Kairouan ***Institut Préparatoire aux Etudes d'Ingénieurs de Kairouan (IPEIK)***	***Série de révision***	***Année universitaire 2017 - 2018*** ***2MP & 2TECHNO & 2PC***

Exercice 1 :

On souhaite écrire le jeu du pendu. Pour rappel, le but du jeu est de trouver un mot caché dont on connaît la taille, en proposant un caractère à chaque essai. Si le caractère proposé existe, on affiche où il se trouve dans le mot (pour toutes ses occurrences) et l'essai n'est pas compté. Si le caractère est incorrect, cela compte pour un essai. L'objectif est de découvrir le mot en un nombre d'essais maximum (5 essais).

Pour écrire ce jeu, on veut utiliser les tableaux de caractères. L'idée est de placer la solution (*i.e.* le mot à trouver) et l'état du jeu dans deux tableaux de caractères. Le tableau de l'état du jeu est de même longueur que la solution mais rempli par le caractère '-'. Lorsqu'un caractère est trouvé, l'état du jeu est mis à jour : on place le caractère trouvé à la place du '-' pour chacune de ses occurrences.

- *En algorithme l'accès à un élément d'une chaîne de caractères peut se faire de la même façon que l'accès à un élément d'un tableau*
- *Les fonctions suivantes sont supposés prédéfinies :*
  - ✓ *long(ch) : retourner la taille de la chaîne de caractères ch.*
  - ✓ *ascii(c) : retourner l'entier correspond au caractère c (c-à-dire le code ascii)*

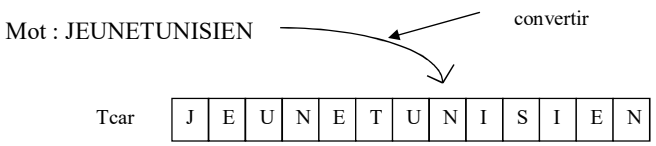
Dans la suite on suppose avoir effectuée les déclarations suivantes :

Type : tab1=tableau [1..50] de caractères

Travail demandé :

1) Ecrivez une fonction algorithmique ***estValide*** prenant en paramètre un mot sous forme d'une chaîne de caractères et retournant *true* si le mot est valide false si non. Un mot est valide si :
   - ✓ Il possède au moins 4 caractères et au maximum 25
   - ✓ Toutes ses lettres sont en majuscule
   - ✓ rappel : les codes ascii obtenus par l'instruction ascii("…") les majuscules sont entre 65 et 90 .

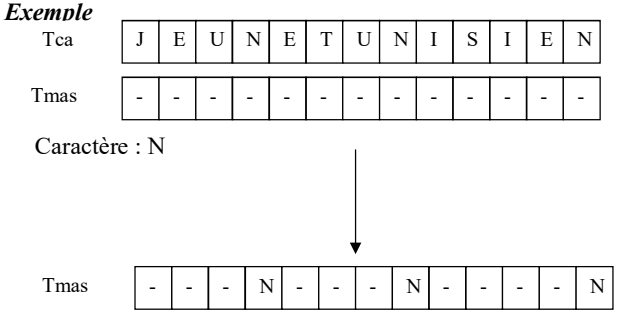
2) Ecrivez la procédure algorithmique ***convertir*** qui prend en paramètre une chaîne de caractères ***Ch*** et un tableau de caractères ***Tcar*** et qui convertit la chaîne ***Ch*** en un tableau de caractères. Chaque case du tableau contient un caractère de la chaîne.



3) Ecrivez la procédure algorithmique ***initialiserSolution*** qui prend en paramètre un tableau de caractères ***Tcar*** et qui demande à l'utilisateur de saisir le mot à trouver et qui remplir ***Tcar*** par ce mot.
   Indications : Vous utiliserez les deux fonction/procédure précédentes (*estValide* et *convertir*)
4) Ecrivez la procédure algorithmique ***creerMasque*** qui remplir le tableau de caractères ***Tmas*** par le caractère '-'. Les paramètres de la procédure ***n, Tmas***

   Exemple n=13

   Tmas | - | - | - | - | - | - | - | - | - | - | - | - | -
5) Ecrivez la procédure algorithmique ***afficher*** prenant en paramètre un tableau de caractères ***Tcar*** et l'affichant à l'écran comme une chaîne de caractères
6) Ecrivez la fonction ***jouer*** prenant en paramètre la solution (tableau de caractères ***Tcar***) , le masque (tableau caractères ***Tmas)*** et un caractère, et retournant ***true*** (si le caractère est présent dans la solution) et mettre à jour le tableau ***Tmas*** : le caractère doit remplacé à la bonne place le caractère '-' (pour toutes ses occurrences). Le cas échéant, retournant ***false***.



7) Ecrivez la fonction ***estFini*** prenant en paramètre le masque ***Tmas*** et qui retourne ***true*** si tous les caractères du mot ont été trouvés et false si non.
8) Ecrivez l'algorithme principal réalisant les actions suivantes :
   a) Demande du mot à trouver à l'utilisateur (procédure initialiserSolution)
   b) Création du masque à partir de la taille de la solution (procédure creerMasque)
   c) Boucle principale du jeu :
      - ✓ Demande à l'utilisateur un caractère

- ✓ Mise-à-jour du masque et du nombre d'essais (suivant si le caractère est présent ou non) (fonction jouer)
- ✓ On s'arrête lorsque **le nombre d'essais atteint 5** ou lorsque le jeu est fini (le mot entier a été trouvé)

d) Affichage du résultat : gagné ou perdu !

Exercice 2

1) Qu'imprime le programme suivant ?

```
a=sorted(list({x//2 for x in range(0,10)}))
b=sorted(list(x//2 for x in range(10,0,-1)}))
c=sorted(list(x//2 for x in range(10,0,-2)]}))
print(a)
print(b)
print(c)
```

2) Soit la fonction mafonction qui retourne $f(x) = x^3 + x^2 + 2x$

```
 #fonctions
Def maFonction(x):
  """Definition d'une fonction de 3eme degre."""
     Return (x**3 + x**2 + 2*x)
```

Écrire la fonction ci-dessus sous forme d'une fonction Lambda

Exercice 3 (les listes) :

1) Ecrire une fonction ***alterner(l,s)*** qui reçoit deux listes de même longueur en argument , et renvoie une liste contenant les éléments des deux argument en alternance, c'est-à-dire ***alterner([1 ,2,3],[4 ,5,6])*** renvoie ***[1,4,2,5,3,6]***
2) Ecrire une fonction ***flatten(l)*** qui reçoit une liste de listes ***l*** en argument, et renvoie la concaténation des sous listes dans une nouvelle liste , par exemple, ***flatten([[2,3,4],[3,5],[1]])*** renvoie ***[2,3,4,3,5,1]***
3) Ecrire une fonction ***diffsym(a,b)*** qui reçoit deux ensemble ***a ,b*** en argument, et renvoie leur différence symétrique, c'est-à-dire leur union sans leur intersection , autrement dit, x est dans l'union symétrique de a et b, si x est dans l'un des deux, mais pas dans les deux en même temps. Pra exemple, ***diffsym({1,2,4,7} ,{1,3,4})*** renvoie ***{2,3,7}.***

Exercice 4: (les dictionnaires)

1) Ecrire une fonction nommée ***dicFreq*** qui reçoit une liste d'entier en argument et retourne le dictionnaire qui associé à chaque chiffre son nombre d'apparition dans la liste.
   Exemple ***dicFreq([1,2,1,3,1,4,1,4])*** retourne ***{1 :4,2 :1,3 :1,4 :2}***

2) En utilisant impérativement ***dicFreq*** écrire une fonction nommée ***dicFreqpour*** qui reçoit une liste d'entier en argument et retourne le dictionnaire qui associé à chaque chiffre son pourcentage d'apparition dans la liste.
   Exemple : ***dicFreqpour([1,2,1,3,1,4,1,4])*** retourne ***{1 :50,2 :12.5,3 :12.5,4 :25}***

NB : les clés peuvent être de n'importe quel type (string , entier, tuple ,…)

Exercice 5

Dans cet exercice, nous considérons une liste de dictionnaires qui représentent une liste d'articles scientifiques. chaque élément de la liste est un dictionnaire du type:

```
{"titre":"Will systems biology offer new holistic paradigms to life sciences?",
"auteur": ("Conti", "Valerio", "Zbilut", "Giuliani"),
"date":2007,
"journal": "Syst Synth Biol.",
"pageDebut": 161,
"pageFin" : 165,
 "volume":1,
"numero": 4}
```

1- Écrire la fonction ***listeTitres*** qui prend en entrée la liste des dictionnaires et retourne la liste des titres des articles publiés.
2- Écrire la fonction ***listeTitres2007*** qui prend en entrée la liste des dictionnaires et retourne la liste des titres des articles publiés en 2007.
3- Écrire une fonction qui prend en argument la liste des dictionnaires et un auteur sous forme de chaîne de caractères et qui renvoie les titres des articles auxquels il a participé.
4- on suppose maintenant que la fonction répondant à la question précédente est écrite. Comment peut-on faire pour avoir la liste des articles auxquels ont participés deux auteurs donnés. Donnez le code de la fonction python correspondante.
5- Écrire une fonction qui prend en argument la liste des dictionnaires et un mot quelconque et qui renvoie la sous listes des dictionnaires qui contiennent ce mot soit dans le titre, soit dans les auteurs,, soit dans le nom du journal.