

	<p align="center"><b>Institut Supérieur des Etudes Technologiques de Sousse</b>  Département de Technologie de l'Informatique  Licence Appliquée en Technologies de l'Informatique  <b>Examen Final</b></p>	<p align="center"><i>Année universitaire : 2021/2022</i>  <i>Semestre : 1</i>  <i>Date : 14 Janvier 2021</i>  <i>Durée : 1h30</i></p>
<p align="center"><i>Unité d'Enseignement :</i>  <b>Option</b></p>	<p align="center"><i>Matière :</i>  <b>Préparation à la Certification Python 1</b></p>	<p align="center"><i>Classes : TI1.*</i></p>
<p><i>Documents :</i>  Non autorisés</p>	<p><i>Enseignants :</i>  A. Elhaj, D. Rouis, G. Benjedidia, M. Ben Jazia, M. Zouari</p>	<p align="center"><i>Nb. Pages : 3</i></p>

### QCM (6 points : 1\*6) :

**Reportez sur votre feuille de réponse le numéro de la question et la/les bonne(s) réponse(s).**

1. Quelle est la sortie du code suivant ?

```
var = 2*[0]+2*[5]
print(var)
```

- a. 10
- b. [0,0],[5,5]
- c. [0, 0, 5, 5]
- d. Un message d'erreur

2. Supposons que t = (1, 2, 4, 4), lesquelles des instructions suivantes génèrent une erreur ?

- a. t.append(10)
- b. print(t[2])
- c. print(len(t))
- d. t[2] = 3

3. Quelle est la sortie du code suivant ?

```
d = {"Mohamed":50498756, "Emna":98556421}
d["Mohamed"]=22214648
print(d)
```

- a. Un message d'erreur
- b. {'Mohamed': 22214648, 'Emna': 98556421}
- c. {"Mohamed" :50498756, "Emna" :98556421, "Mohamed":22214648}
- d. {"Mohamed":50498756, "Emna":98556421}

4. Quelle est la sortie du code suivant ?

```
x={}
x[2]=1
x[1]=[2,3,4]
print(x[1][1])
```

- a. 4
- b. 3
- c. [2,3,4]
- d. Un message d'erreur

5. Une clé dans un dictionnaire peut être :

- a. Une chaîne de caractères
- b. Un entier
- c. Une liste
- d. Un tuple

6. Quelle est la sortie du code suivant ?

```
my_list = [1, 2, 3, 4, 5]
del my_list[0:2]
print(my_list)
```

- a. [1, 2, 3]
- b. [4, 5]
- c. [3, 4, 5]
- d. Un message d'erreur

### Exercice 1 (4 Pts)

Ecrire un programme python qui permet de saisir un nombre n strictement positif puis affiche un triangle formé par n lignes de chiffres identiques comme dans cet exemple (pour n=7) :

```
7777777
666666
55555
4444
333
22
1
```

### Exercice 2 (10 points : 1.5+1.5+1.5+2+2+1.5)

Le but de cet exercice est d'écrire un programme python pour gérer certaines données du réseau social « Twitter ». Les données ci-après (utilisant la structure de **dictionnaire Python**) représentent un extrait de l'entité **Tweet** (Un Tweet représente un message posté par un utilisateur du site) :

```
{
    "created_at": "01/01/2022",
    "id": 1050118621198921720,
    "text": "Happy New Year",
    "user": {
        "id": 6253282,
        "name": "Bertrand",
    }
}
```

D'après cet extrait, on peut comprendre que l'utilisateur N°6253282 dont le nom est « Bertrand » a posté le 01/01/2022 un tweet contenant le message « Happy New Year ». Ce tweet a pour numéro : 1050118621198921720.

Tous les tweets postés sont regroupés dans une liste **Tweets**.

1. Ecrire une fonction python **add\_tweet(tweets)** qui prend en paramètre une liste de tweets et qui permet de lui ajouter un nouveau Tweet en Effectuant les opérations suivantes :
  - **Faire appel** à une fonction supposée existante **init\_tweet()** qui retourne un tweet initialisé avec l'id, la date et les informations (id et nom) du user (on ne vous demande pas d'écrire le code de **init\_tweet()**) ;
  - **Saisir** le message (le "text") du tweet qu'on vient d'initialiser ;
  - **Ajouter** le Tweet résultant à la fin de la liste tweets.

2. Ecrire une fonction python **user\_total\_tweets(id,tweets)** qui prend en paramètre une liste de tweets et un id d'utilisateur et qui retourne le nombre des tweets contenus dans cette liste et appartenant à cet utilisateur.
3. Ecrire une fonction python **search\_msg(ch,tweets)** qui prend en paramètre une chaîne « ch » et une liste de tweet « tweets » et qui permet d'afficher tous les tweets (texte du message uniquement) de cette liste contenant ch. Si ch est vide, la fonction n'affiche rien.

**NB.** L'opérateur « **in** » permet de vérifier si une chaîne est contenue dans une autre (l'expression (**ch1 in ch2**) est évaluée à **True** si ch1 est une sous-chaîne de ch2).

4. Ecrire une fonction python **set\_tags(tweet,n)** qui prend en paramètre un tweet et un entier n (supposé >0), et qui permet d'ajouter une clé « **tags** » à ce tweet, avec comme valeur associée à cette clé une liste contenant n chaînes de caractères saisies par l'utilisateur dans la fonction et devant commencer par le symbole « **#** ».

**Par exemple**, pour n égal à 3, si l'utilisateur saisit les chaînes "#Tunisie", "#Test" et "#tweet", le tweet passé en argument est mis à jour en ajoutant la clé "tags" associée à la valeur ["#Tunisie", "#Test", "#tweet"] ce qui donne comme résultat :

```
{
    "created_at": "01/01/2022",
    # Autres clés ici (id, text, user)
    "tags" : ["#Tunisie", "#Test", "#tweet"]
}
```

5. Ecrire une fonction python **del\_tweet(id,tweets)** qui prend en argument id un identifiant d'un tweet et une liste de tweets et qui permet de supprimer ce tweet de la liste. La fonction retourne **True** si un tel tweet a été trouvé dans la liste (et donc supprimé) et **False** sinon (cas où le id est inexistant). La valeur de l'id est unique.
6. Ecrire le **programme principal** qui permet d'effectuer les opérations suivantes (en **faisant appel aux fonctions adéquates**) :
  - Remplir la liste « Tweets » par n « tweet », n étant un entier **strictement positif** donné par l'utilisateur ;
  - Saisir un mot (**chaîne non vide**) au clavier puis afficher tous les textes des « tweets » qui contiennent le mot saisi.