2020-2021
Dr. SECK

TP PYTHON L3 TDSI (MATHS-CRYPTO)

1 Exercice 1 :

0. Ecrire un code python qui demande à l'utilisateur une chaine de caractères appelée `chaine` et affiche `chaine` en insérant un tiret `-` comme suit:

   - si le nombre de caractères dans `chaine` est `impair`, on remplace le caratère au milieu (de `chaine`) par `-`.
   - si le nombre de caractères dans `chaine` est `pair`, on insère `-` au milieu (de `chaine`).

   Par exempe:

   - si `chaine = "BOnjour"` (7 caratères), on affiche `"BOn-our"`
   - si `chaine = "L'equipe TDSI"` (13 caratères), on affiche `"L'equi-e TDSI"`
   - si `chaine = "A"` (1 caratère), on affiche `"-"`
   - si `chaine = "TDSI"` (4 caratères), on affiche `"TD-SI"`

1. Ecrie un code python qui dit si un entier `p` donné par l'utilisateur est premier ou non. On rapelle qu'un nombre premier est un entier qui n'est divisible que par `1` et lui-même.

2. Ecrie un code python qui affiche le type d'une variable `var` saisie par l'utilisateur. On pourra utiliser les fonctions `eval` et `isinstance` (ou `type`). Par exemple si

   - `var = 1`, on afficher: "Le type de var es `int`"
   - `var = {1: 2, "A": 0}`, on afficher: "Le type de var es `dict`"

3. Ecrire une condition `if` pour vérifier si une variable appelée `test1` est supérieure à `5`. Si c'est le cas, afficher "Félicitations".

4. Ecrire une condition `if` pour vérifier si une variable appelée `test2` est impaire. Si elle l'est, affichez "J'ai trouvé un nombre impair" ; sinon, affichez "J'ai trouvé un nombre pair".

5. Ecrire une condition `if` pour vérifier si une variable appelée `test3` n'est pas `entier`. Si ce n'est pas le cas, affichez "Un nombre fantastique". Sinon, ne faites rien.

6. Ecrire une condition `if` pour vérifier si une variable appelée `test4` est dans l'intervalle `[-3, 3]`. Si c'est le cas, affichez "Petit nombre !" Si elle est en dehors de cet intervalle mais dans l'intervalle `[-10, 10]`, affichez "Nombre moyen !"; et si elle est en dehors de cet intervalle, affichez "Grand nombre !"

7. Ecrire une condition `if` pour vérifier si une variable appelée `test6` est un entier négatif. Si c'est le cas, multipliez-la par `-3` et voyez si le résultat est un multiple de `9` ; si c'est le cas, affichez "hello, mon ami". Dans tous les autres cas, affichez "Oh, non, pas vous".

8. Ecrire un programme qui lit le numéro du jour de la semaine (de `1` à `7`). S'il s'agit d'un jour ouvrable (Lundi à vendredi), le programme écrira le nom du jour correspondant. Sinon, il écrira le mot "Week-end".

9. Faire un programme qui, en donnant un numéro de mois (de `1` à `12`), indique le nombre de jours dont il dispose `(28, 30 ou 31)`, en ignorant les années bissextiles.

10. En utilisant une boucle `for`, ecrire un code python qui calcule la somme des carrés des `n` premiers entiers naturels pour `n` donné par l'utlisateur i.e

$$1^2 + 2^2 + 3^2 + \ldots + n^2$$

11. Lire un nombre entier `n` entre `0` et `9` et afficher sa table de multiplication jusqu'à `N` où `N` est un autre entier naturel lu par le programme.

12. Ecrire un programme qui affiche tous les entiers nuturels entre `0` et `40` qui sont des multiples de `3, 7` ou `11`.

13. Calculer les expressions suivantes pour un nombre naturel `n` choisi par l'utilisateur :

(i). $\displaystyle\sum_{i=0,\ i \text{ pair}}^{n-1} (i+3)$

(ii). $\displaystyle\prod_{j=0,\ j \text{ multiple de } 3}^{n-1} (j^2 + 2j)$

14. Si on énumère tous les nombres naturels inférieurs à 10 qui sont des multiples de `3` ou `5`, on obtient `3, 5, 6` et `9`. La somme de ces multiples est `23`. Ecrire un code en python qui affiche la liste et la somme de tous les multiples de `3` ou `5` en inférieurs à `200`.

2 Exercice 2 : Chiffrement par décalage

On pose

```
alphabet= "ABCDEFGHIGKLMNOPQRSTUVWXYZ"
```

2.1 Partie 1:

1. Definir un dictionnaire appelé `dico` qui identifie une lettre de l'aphabet à sa position: $A \to 0; B \to 1, \ldots Z \to 25$

2. Definir un dictionnaire appelé `dico_inv` qui identifie un entier `n` compris entre `0` et `25` à la lettre de l'aphabet qui est à la position `n` : $0 \to A; 1 \to B, \ldots 25 \to Z$

3. Demander à l'utilisateur de saisir un message à crypter (la variable sera appelée `message`). Ensuite définir une variable `caract_message` qui sera la liste des caractères de `message` en majuscule. On enlèvera tous les accents. On ignorera également tout caractère qui n'est pas dans `alphabet`. Par exemple

- si `message = "C'est très confidentiel"`, alors `caract_message` sera égale à `['C', 'E', 'S', 'T', 'T', 'R', 'E', 'S', 'C', 'O','N', 'F', 'I', 'D', 'E', 'N', 'T', 'I', 'E', 'L']`
- si `message = "OH! Soldat, ne tirer pas"` alors `caract_message` sera égale à `['O', 'H','S', 'O', 'L', 'D', 'A', 'T', 'N', 'E', 'T', 'I', 'R', 'E', 'R', 'P', 'A', 'S']`

4. Demander à l'utilisateur de donner une clef $K \in \{0, 1, 2, \ldots, 25\}$‘
5. Chiffrer chaque lettre de `caract_message` et regroupez les tous sous forme d'une liste dans une variable appelée `chiffre` en utilisant le chiffrement par décalage.
6. Afficher le message chiffré sous forme d'une chaine de cractères qui sera la concaténation de tous les caractères chiffrés.

2.2 Partie 2:

Proposer un programme en python qui permet de déchiffrer un message chiffré (par la méthode de décalage)

3 Exercice 3 : Recherche

1. Donner quatre opérations qu'on peut faire sur variable de type `dict`. On donnera à chaque fois des exemples.
2. Comment fonctionne la boucle `while` en python ? Faire une comparaison entre les boucles `for` et `while`. On donnera à chaque fois des exemples.
3. Que fait les mot-clefs `continue, break` dans une boucle
4. Expliquer brièvement ce que fait les fonctions suivantes `enumerate, zip, min, max, sum, sorted`