

**Nom : DANIEL KIHIMBA KULE**

**PROMOTION : MASTER PRO IRSS**

**REPONSES DE L'EXAMEN DE PROGRAMMATION PYTHON**

1. **R/** Un algorithme est une suite d'instructions définies et ordonnée conçue pour résoudre un problème.
2. **R/** Non, en python, il n'est pas nécessaire de déclarer le type d'une variable avant de lui attribuer une valeur. Python est un langage à typage dynamique, le type d'une variable est déterminé automatiquement en fonction de la valeur qui lui est assignée.
3. **R/** - **range()** : permet de générer une séquence d'un point X à un point y.
  - **Int()** : permet de convertir une valeur en nombre entier.
  - **Zip()** : permet de regrouper plusieurs listes, tuples, ... en un seul itérable en associant tous les éléments de même indice.
4. **R/** **Lecture** : `open( ), file.read( ), file.readlines()`,  
**Ecriture** : `open(nom fichier, 'w'), file.write( ), file.writelines( )`

**PARTIE II**

**EXERCICE 1**

**Reponses :**

1. `type(11.7)` : retourne le type de la valeur 11.7
2. `L = (2,2,2,24)` : crée un tuple L contenant les valeurs (2,2,24)  
`L = [2]` : accède à l'élément à indice 2 du tuple L qui est 24/
3. `L = (L,5)` : crée un nouveau tuple L contenant la valeur 5  
`L = (1)` : Accède à l'élément à l'indice du tuple L qui est 5
4. `tuple(range(10, 1, -2))` : crée un tuple de séquence d'entiers générée par `range( 10, 1, -2)`
5. `list(range(6))` : crée une liste contenant des entiers de 0 à 5.
6. `Not(True or False)` : Evalue l'expression booléenne True or False, qui, est True, puis nie le résultat ce qui donne false.
7. `Ch = "Master Professionnel 2024/2025"` :  
`Len(ch)` : retourne la longueur de la chaîne stockée dans la variable

8. `Ch.find("1")` : recherche la première occurrence de la chaîne "1" dans la variable `ch`.
9. `Ch.count("m")` : compte le nombre d'occurrence de la chaîne "m" dans la variable `ch`.
10. `Set(list(ch[:20]))` : crée un (set) à partir des 20 premiers caractères de la variable `ch`.
11. `11 != 7` : effectue la comparaison ; au cas où c'est différent de, il va retourner `True`.

## EXERCICE 2

```
R/ score = float(input("Entrez un score entre 0,0 et 1,0 : "))
if score < 0.0 or score > 1.0:
    print("Erreur : le score doit être compris entre 0,0 et 1,0.")
elif score >= 0.9:
    print("Note : A")
elif score >= 0.8:
    print("Note : B")
elif score >= 0.7:
    print("Note : C")
else:
    print("Note : F")
```

## EXERCICE 3

```
R/ def add_dic(N) :
    result = {}
    for w in range(1, N+1):
        result[w] = w + w
    return result
```