

NOM Prénom :

1. Donner la valeur contenue dans la variable `val` après exécution et le type python du résultat (avec une brève explication si besoin).

instruction	valeur	type (python)
<code>val = 7//4</code>	1 (quotient de la division de 7 par 4 : $7 = 1 \times 4 + 3$)	<code>int</code>
<code>val = 7 % 5</code>	2 (reste de la division de 7 par 5)	<code>int</code>
<code>val = 2**3</code>	8 (2^3)	<code>int</code>
<code>val = '2'+'3'</code>	'23' (concaténation de '2' et '3')	<code>str</code>
<code>val = (5+4 < 6)</code>	False ($9 < 6$ est faux)	<code>bool</code>
<code>val = 3*[0,2]</code>	[0, 2, 0, 2, 0, 2]	<code>list</code>

2. Donner une expression booléenne testant si l'entier contenu dans la variable `n`

a) est un multiple de 7 : `m % 7 == 0`

b) est un multiple de 5 inférieur ou égal à 2028 : `(m % 5 == 0) and (m <= 2028)`

3. On considère la liste `L = [2, 3, 3, 7, 5]`.

a) Écrire l'instruction permettant d'obtenir la longueur de la liste `L` : `len(L)`

b) Rappeler précisément la plage des indices admissibles pour une liste `M` quelconque : :

`[0; p-1]` ou `[0; p[` = les entiers de 0 inclus à `p` (taille de `M`) exclu

c) On exécute l'instruction suivante : `L[2]=7`

Donner alors la liste `L` obtenue et sa longueur `[2, 3, 7, 7, 5]` longueur 5

4. Donner l'instruction permettant d'initialiser la liste `V` à la liste des valeurs de $5k - 3$ pour k dans `[2; 14]`. On utilisera une définition par compréhension.

`V = [5*k - 3 for k in range(2, 14+1)]`

5. On considère le script

```
texte = 'port'
n = len(texte)
code=""
for i in range(0,n) :
    caract = texte[i]
    code = caract + code
print(code)
```

Détailler le fonctionnement pas à pas de la validation de ce script : on donnera en particulier les valeurs successives prises par les différentes variables

La variable texte prend la valeur 'port' (str) puis $n=4$
et code = ""
Pour $i=0$ caract='p' donc code='p'
Pour $i=1$ caract='o' donc code='op'
Pour $i=2$ caract='r' donc code='rop'
Pour $i=3$ caract='t' donc code='trop'
On sort de la boucle (i varie de 0 inclus à 4 exclu)
On affiche le contenu de code soit trop

6. On suppose validé le contenu d'une liste non vide d'entiers stockée dans la variable Li.

a) Écrire un script (suite d'instructions) permettant de calculer la somme des nombres de la liste Li et de l'affecter à la variable totalite.

```
totalite = 0
for k in range(0, len(Li)):
    totalite = totalite + Li[k]
# ici totalite a la valeur attendue

ou par un parcours par valeurs
totalite = 0
for elt in Li:
    totalite = totalite + elt
# ici totalite a la valeur attendue
```

NOM Prénom :

b) Écrire un script affectant à la variable entre le nombre d'éléments de la liste Li compris strictement entre 7 et 16.

```
entre = 0
for elt in Li:
    if (elt > 7) and (elt < 16):
        entre = entre + 1
print(entre)
```

c) Écrire un script affectant à la variable quad la liste des éléments de la liste Li, multiples de 4.

```
quad = []
for i in range(0, len(Li)):
    elt = Li[i]
    if elt % 4 == 0: # elt multiple de 4
        quad.append(elt)
# ici quad contient ce qui est attendu
```

7. On définit une suite par récurrence par :

$$u_0 = -2 \quad \text{et} \quad \forall n \in \mathbb{N}, \quad u_{n+1} = \begin{cases} 2 \times u_n & \text{si } n \text{ multiple de } 3 \\ -u_n + 10 & \text{si } n \text{ vaut du } 3k + 2 \\ u_n & \text{sinon} \end{cases}$$

Écrire un script qui fabrique la liste des éléments de la suite de l'indice 0 à 2027 compris.

```
L = [-2] # liste [u0]
u = -2
for m in range(0, 2027): # dernière valeur 2026
    # calcul de u_{m+1}
    if m % 3 == 0:
        L.append(2 * L[m])
    elif m % 3 == 2:
        L.append(10 - L[m//3])
    else:
        L.append(L[m])
```

8. On suppose que la variable Lint contient une liste d'entiers. Écrire un script qui détermine le minimum des éléments de cette liste et un indice où il est atteint.

```
mini = Lint[0]
ind = 0
for k in range(0, len(Lint)): # on peut commencer à 1
    if Lint[k] < mini:
        mini = Lint[k]
        ind = k

# ici mini contient le minimum et ind le plus petit indice où ce minimum est atteint.
```