

DS bases de la programmation

Durée de l'épreuve : **01h50**

L'usage de la calculatrice n'est pas autorisé.

Le candidat répond sur feuilles doubles numérotées et garde l'énoncé.

Les traces de recherche, même incomplètes ou infructueuses, seront valorisées.

La qualité de la rédaction, la clarté et la précision des raisonnements seront prises en compte.

Exercice 1 (10 points)

1. Valeur de variables (1 pt)

Déterminer la valeur des variables **a**, **b**, **c** et **d** après exécution du programme ci-dessous :

```
a = 3
b = a + 2
a = a + 2
c = a + b
a = b
a = a + 1
b = c % b
d = c - 6 / a

print(f'a = {a} , b = {b} , c = {c} , d = {d}')
```

2. Listes (1 pt)

Déterminer la valeur des variables **a**, **b**, **c** et **d** après exécution du programme ci-dessous :

```
liste_1 = [4, 3, 2, 1]
a = liste_1[2]
liste_2 = [1, 2, 3, 4]
liste_3 = [liste_1, liste_2, liste_1]
b = liste_3[1]
c = liste_3[1][2]
d = b[3]

print(f'a = {a} , b = {b} , c = {c} , d = {d}')
```

3. Parité (2 pts)

Écrire un programme qui demande à l'utilisateur deux nombres entiers et qui lui affiche :

- **"somme paire"** si la somme de ces deux entiers est paire ;
- **"somme impaire"** si la somme de ces deux entiers est impaire.

4. Attention (2 pts)

Écrire un programme optimisé qui demande à l'utilisateur de saisir son pourcentage d'attention en cours de NSI sous forme d'un nombre entier entre 0 et 100 :

- si le nombre entier renseigné par l'utilisateur n'est pas entre 0 et 100 , on lui demandera de resaisir un nombre, et cela autant de fois que nécessaire ;
- si ce nombre est inférieur à 50, on lui affichera le message : **"Tu dors ou quoi !"** ;
- si ce nombre est entre 50 et 80, on lui affichera le message **"Concentre toi davantage !"** ;
- si ce nombre est supérieur à 80, on lui affichera le message **"Bravo, continue ainsi !"**.

NB : on suppose que le nombre renseigné est bien un nombre entier, mais pas forcément entre 0 et 100.

5. Triangle (2 pts)

Écrire une fonction **triangle**, qui prend en paramètre une longueur **taille (int)** et qui, à l'aide du module **turtle**, trace un triangle équilatéral de côté de longueur **taille**.

```
def triangle(taille: int):  
    pass
```

6. Images (2 pts) Écrire une fonction **images** qui prend en paramètre une liste d'entiers **tab** et renvoie la liste de leur carré.

```
def images(tab: list) -> list:  
    pass  
  
assert images([0, -1, 1, 2, -2, 3, 10]) == [0, 1, 1, 4, 4, 9, 100]  
assert images([ ]) == [ ]
```

Exercice 2 (5 points)

On ne peut pas utiliser dans cet exercice la méthode Python **index**.

1. Recopier et compléter la fonction *myIndex* qui :

- prend en paramètres un entier **n** et une liste d'entiers **tab** ;
- si **n** n'est pas dans **tab** : renvoie -1 ;
- si **n** est dans **tab** : renvoie l'index de la première occurrence de **n** dans **tab** ;
- utilise une boucle **for**.

```
def myIndex(n: int, tab: list) -> int:  
    for i in range(.....):  
        if .....:  
            .....  
        .....  
    .....  
  
myIndex(1, [0, 0, 1, 0, 1]) == 2  
myIndex(1, [0, 0, 0, 0, 1]) == 4  
myIndex(3, [0, 0, 1, 0, 1]) == -1
```

2. Écrire une fonction *myLastIndex* qui :

- prend en paramètres un entier **n** et une liste d'entiers **tab** ;
- si **n** n'est pas dans **tab** : renvoie -1 ;
- si **n** est dans **tab** : renvoie l'index de la dernière occurrence de **n** dans **tab** ;
- utilise une boucle **while**.

```
def myLastIndex(n: int, tab: list) -> int:  
    pass  
  
myLastIndex(1, [0, 0, 1, 0, 1]) == 4  
myLastIndex(1, [1, 0, 1, 0, 0]) == 2  
myLastIndex(3, [0, 0, 1, 0, 1]) == -1
```

Exercice 3 (5 points)

Deux joueurs s'affrontent au jeu du 21.

On commence à zéro ; puis chaque joueur ajoute alternativement 1, 2 ou 3 ; le premier qui atteint 21, ou plus, perd.

Par exemple, avec cette séquence le joueur 1 perd :

Joueur 1		3		1		2		3		1		1 (ou 2 ou 3)
Joueur 2			2		3		1		1		3	
Somme	0	3	5	6	9	11	12	15	16	17	20	21 (22 ou 23)

Ecrire une fonction `jeu21` qui :

- prend en paramètres deux noms de joueurs ;
- joue au jeu du 21 en demandant alternativement aux deux joueurs leur nombre parmi 1, 2 ou 3 ;
- renvoie le nom du joueur perdant.

```
def jeu21(nomJoueur1: str, nomJoueur2: str) -> str:
    pass

nomPerdant = jeu21("Turing", "Al-khawarizmi")
print(f'{nomPerdant} a perdu')
```

```
Turing, donne ton choix (1, 2 ou 3) : 3
somme = 3
Al-khawarizmi, donne ton choix (1, 2 ou 3) : 2
somme = 5
Turing, donne ton choix (1, 2 ou 3) : 1
somme = 6
Al-khawarizmi, donne ton choix (1, 2 ou 3) : 3
somme = 9
Turing, donne ton choix (1, 2 ou 3) : 2
somme = 11
Al-khawarizmi, donne ton choix (1, 2 ou 3) : 1
somme = 12
Turing, donne ton choix (1, 2 ou 3) : 3
somme = 15
Al-khawarizmi, donne ton choix (1, 2 ou 3) : 1
somme = 16
Turing, donne ton choix (1, 2 ou 3) : 1
somme = 17
Al-khawarizmi, donne ton choix (1, 2 ou 3) : 3
somme = 20
Turing, donne ton choix (1, 2 ou 3) : 1
somme = 21
Turing a perdu
```