

DS bases de la programmation

Durée de l'épreuve : **01h30**

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. Type de variables (1 pt)

Quels sont les types de données pour les variables a, b, c, d définies dans le programme ci-dessous :

```
a = 2.3 + 3.7      # Type a : .....
b = 6 / 3          # Type b : .....
c = 2 % 4          # Type c : .....
d = "Test" == "1" # Type d : .....
```

2. Âge limite (1pt)

Écrire un programme qui demande à l'utilisateur d'entrer son âge et affiche le message :

- o "Autorisé" si l'âge saisi est supérieur ou égal à 16
- o "Non autorisé" sinon

3. Nombres entiers entre 5 et 200 (2 pts)

On souhaite afficher tous les nombres entiers de 5 à 200 (5 et 200 compris).

- a. Écrire un premier programme qui utilise une boucle for.
- b. Écrire un second programme mais cette fois-ci, en affichant les nombres par ordre décroissant et en utilisant une boucle while.

4. Syracuse (2 pts)

- a. Que contiendra la variable **t** à la fin de l'exécution du script Python ci-dessous ? (1 pt)

```
def syracuse(x):
    if x % 2 == 0:
        return x/2
    return 3*x+1

t = syracuse(5)
```

- b. On rajoute au programme précédent les instructions suivantes :

```
t = 4
while t > 1:
    t = syracuse(t)
    print(t)
```

Quel affichage obtient-on ? (1 pt)

5. Économiser une somme d'argent annuellement (2 pts)

Écrire une fonction, nommée *economiser()* qui à partir des deux paramètres *m* (montant, un entier positif) et *a* (nbre d'années, un entier positif) :

- affiche année par année, le message : "Montant économisé : ", suivi du montant cumulé
- puis retourne le montant accumulé au bout des *a* années

```
>>> print("Montant final : ", economiser(10000, 5))
Montant économisé : 10000
Montant économisé : 20000
Montant économisé : 30000
Montant économisé : 40000
Montant économisé : 50000
Montant final : 50000
```

6. Saisie utilisateur correcte (2 pts)

Écrire une fonction *validerSaisie()* qui :

- reçoit un paramètre *v*, de type entier, supérieur à 0,
- demande et redemande le cas échéant à un utilisateur d'entrer une valeur numérique quelconque jusqu'à ce que la valeur entrée soit strictement supérieur à *v*,
- puis retourne la valeur saisie par l'utilisateur.

Exercice 2 - 1ère spé NSI 3 (3 points)

Écrire une fonction *moyenne(n)* qui calcule et retourne la moyenne de *n* entiers aléatoires entre 5 et 20.

Aucune fonction prédéfinie de Python n'est autorisée à part *randint()* de la bibliothèque *random*.

Rappel de l'utilisation de la fonction *randint()* :

l'instruction *randint(2,5)* retourne une valeur aléatoire entière comprise entre 2 et 5 inclus.

Exercice 2 - 1ère spé NSI 6 (3 points)

Écrire une fonction *moyenneListe()* qui prend en paramètre une liste d'entiers et renvoie la moyenne des éléments de la liste :

```
>>> print(moyenneListe([10, 12, 14, 16]))
13
```

Exercice 3 (7 points)

On va programmer une version très simplifiée du jeu 007.

Chacun des deux joueurs choisit un coup parmi :

- recharger son pistolet ("**R**"),
- se protéger d'un tir de pistolet ("**SP**"),
- tirer avec son pistolet ("**T**").

Le joueur n'a pas besoin de recharger au préalable son pistolet avant de tirer pour cette version simplifiée.

Quand il y a égalité, les deux joueurs rejouent jusqu'à que l'un deux remporte la manche.

La partie est remportée par le premier joueur qui gagne trois manches.

1. Implémenter une fonction `inputJoueur()` qui demande à l'utilisateur son coup et le retourne : (1,5 pts)

```
>>> print("coup joueur 1 : ", inputJoueur())
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
coup joueur 1 : R
>>> print("coup joueur 2 : ", inputJoueur())
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : SP
coup joueur 2 : SP
```

2. Recopier et compléter la fonction `resultatCoup()` qui : (1,5 pts)

- prend en paramètres le coup du joueur 1 et le coup du coup du joueur 2,
- retourne 1 si le coup du joueur 1 bat le coup du joueur 2, 2 si le coup du joueur 2 bat le coup du joueur 1 et 0 s'il y a égalité.

```
def resultatCoup(coup_joueur_1, XXXXX):
    """
    Renvoie :
        0 si égalité
        1 si coup_joueur_1 bat coup_joueur_2
        2 si coup_joueur_2 bat coup_joueur_2
    """
    if coup_joueur_1 == XXXXX and XXXXX:
        return XXXXX
    elif XXXXX:
        return XXXXX
    XXXXX:
        return XXXXX
```

```
>>> print(resultatCoup('T', 'T'))
0
>>> print(resultatCoup('T', 'SP'))
0
>>> print(resultatCoup('T', 'R'))
1
>>> print(resultatCoup('R', 'T'))
2
```

3. Implémenter une fonction *mancheJeu007()* qui :

(2 pts)

- demande aux deux joueurs leur coup jusqu'à que l'un des deux gagne,
- retourne 1 si le joueur 1 gagne la manche et 2 si le joueur 2 gagne la manche.

```
>>> print("Le vainqueur de la manche est le joueur ", mancheJeu007())
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Résultat coup : 0
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Résultat coup : 0
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Résultat coup : 2
Le vainqueur de la manche est le joueur 2
```

4. Implémenter une fonction *partieJeu007()* qui :

(2 pts)

- lance plusieurs manches jusqu'à que l'un des deux joueurs gagne trois manches et donc la partie,
- retourne 1 si le joueur 1 gagne la partie et 2 si le joueur 2 gagne la partie.

```
>>> print("Le vainqueur de la partie à trois points est le joueur ", partieJeu007())
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Résultat coup : 1
Le vainqueur de la manche est le joueur 1
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Résultat coup : 2
Le vainqueur de la manche est le joueur 2
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Résultat coup : 1
Le vainqueur de la manche est le joueur 1
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : T
Votre coup ("R" pour recharger, "SP" pour se protéger, "T" pour tirer) : R
Résultat coup : 1
Le vainqueur de la manche est le joueur 1
Le vainqueur de la partie à trois points est le joueur 1
```

Exercice bonus (optionnel)

Implémenter le jeu 007 avec les règles supplémentaires suivantes :

- on ne peut tirer au pistolet ("*TP*") que si on a au moins une recharge pistolet ("*RP*"), le nombre de recharge se décrémentant à chaque fois qu'on se protège d'un tir pistolet ("*RP*"),
- au bout d'au moins trois recharges Kamé Hamé Ha ("*RK*"), on peut faire un tir Kamé Ha Mé ("*TK*") contre lequel on ne peut pas se protéger.