

## Examen Final

Durée : 2 heures

Calculatrices, téléphones, traducteurs et ordinateurs interdits.

Feuille de notes A4 recto-verso autorisée

**Attention : chaque partie doit être rédigée sur une copie séparée**

N.B. : on s'attachera à fournir tout commentaire utile et à écrire de façon claire et lisible.

### 1<sup>ère</sup> partie : Fonctions et procédures (8 points)

#### 1. Généralités (3 points)

Que s'affiche-t-il lors de l'exécution des fonctions suivantes ?

<p>a)</p> <pre>x = 5 def fcn1(x):     return x**2 print(fcn1(2), x)</pre>	<p>b)</p> <pre>a = 5 def fcn2(x):     a = 2     return a+x print(fcn2(2), a)</pre>
<p>c)</p> <pre>a = 5 def fcn3(x):     a += 1     return a+x print(fcn3(2), a)</pre>	<p>d)</p> <pre>def fcn4(x, a, b):     return a*x + b print(fcn4(2, 3))</pre>
<p>e)</p> <pre>a = 5 def fcn5(x, a = 6, b = 3):     a = 4 + a     return a*x + b print(fcn5(2, b = 3))</pre>	<p>f)</p> <pre>def fcn6(x, a = 6, b = 3):     return a*x + b print(fcn6(2, b = 3, a = -3))</pre>

#### 2. Coffee break (5 pts)

On se propose de réaliser une version (très simplifiée) d'un distributeur de café. Le client doit choisir un produit, dans la liste : espresso (€2,00), café allongé (€1,50) et café au lait (€2,50). Le distributeur doit afficher le prix à payer, en suite recevoir le paiement en pièces et restituer la monnaie le cas échéant :

- A) Ecrire une fonction `prix` qui affiche le prix à payer selon le type de produit qu'elle reçoit en paramètre (1.5 point)
- B) Ecrire une fonction `paiement` qui encaisse le prix à payer. On attend de cette procédure qu'elle affiche, aussi longtemps que nécessaire, la somme qui reste à payer, et demande au client d'introduire une pièce autorisée (0,1€ ; 0,2€ ; 0,5€ ; 1€ ; 2€). Une fois le paiement effectué, elle doit afficher la monnaie à reverser au client et le renvoyer au programme principal. (2 points)
- C) Présentez le programme principal qui sollicite cette fonction et cette procédure. (1.5 points)

----- { *prendre une nouvelle copie* } -----

## 2<sup>ème</sup> partie : Autres (13 points)

1. Un pangramme est une phrase comportant au moins une fois chaque lettre de l'alphabet. Transformer cette fonction itérative en fonction récursive (1.5 points) ?

```
def pangramme_ite(s):
    pangramme = True
    alphabet = "abcdefghijklmnopqrstuvwxy"
    i = 0
    while i < len(alphabet) and pangramme:
        if not alphabet[i] in s:
            pangramme = False
        else :
            i += 1
    return pangramme
```

2. Quel est le type de `p` et qu'affiche le `print` ? (1 point)

```
p = {"Dali", "Pollock", "Magritte", "Basquiat"}
p.update(["Calder", "Basquiat"])
p.add("Calder")
print(p)
```

3. Proposer une fonction Python qui permute la lettre se trouvant à l'indice `i` avec la lettre se trouvant à l'indice `j`. Il est interdit de passer par un tableau (1.5 points).

Par exemple, `permuter("Men artiste préféré est Basquiat", 1, 10)` renvoie "Mon artiste préféré est Basquiat"

4. Imaginons le tableau de dictionnaires suivant, où chaque dictionnaire représente un film.

```
films = [{"nom" : "Harry Potter à l'école des sorciers", "genre" :
          "Fantastique", "date" : 2001, "duree" : 152},
         {"nom" : "Charlie et la chocolaterie", "genre" :
          "Fantastique", "date" : 2005, "duree" : 112},
         {"nom" : "Les gardiens de la galaxie", "genre" :
          "SF", "date" : 2014, "duree" : 121},
         {"nom" : "The social network", "genre" :
          "Drame", "date" : 2010, "duree" : 120},
         {"nom" : "Interstellar", "genre" :
          "SF", "date" : 2014, "duree" : 169},
         {"nom" : "Le Fabuleux Destin d'Amélie Poulain", "genre" :
          "Comédie romantique", "date" : 2001, "duree" : 122},
         {"nom" : "Monty PYTHON : Sacré Graal", "genre" :
          "Fantastique", "date" : 1975, "duree" : 91}]
```

Proposer une fonction Python `defNotes` qui demande à des amis de noter chaque film avec une note allant de 1 à 5. On souhaite ajouter au tableau `films` la somme de ces notes pour chaque film. (2.5 points)

Par exemple `films = defNotes(films, ["Morgan", "Myriem", "Salomé", "Yasmine", "Yassine"])` pourrait renvoyer, la liste suivante, si tout le monde a mis 5/5 à Harry Potter, 4.5/5 à Charlie et la chocolaterie, etc. :

```
films = [{"nom" : "Harry Potter à l'école des sorciers", "genre" :
          "Fantastique", "date" : 2001, "duree" : 152, "note" : 25},
         {"nom" : "Charlie et la chocolaterie", "genre" :
          "Fantastique", "date" : 2005, "duree" : 112, "note" : 22.5},
         {"nom" : "Les gardiens de la galaxie", "genre" :
          "SF", "date" : 2014, "duree" : 121, "note" : 20},
         {"nom" : "The social network", "genre" :
          "Drame", "date" : 2010, "duree" : 120, "note" : 17.5},
         {"nom" : "Interstellar", "genre" :
          "SF", "date" : 2014, "duree" : 169, "note" : 15},
         {"nom" : "Le Fabuleux Destin d'Amélie Poulain", "genre" : "Comédie
          romantique", "date" : 2001, "duree" : 122, "note" : 12.5},
         {"nom" : "Monty PYTHON : Sacré Graal", "genre" : "Fantastique",
          "date" : 1975, "duree" : 91, "note" : 10}]
```

5. Quel est le type de tri utilisé dans la fonction suivante ? Donner un exemple d'appel et de retour avec le dictionnaire `film` présenté ci-dessus. (1.5 points)

```
def tri_XXX(tinit, cledico1, cledico2, valeurdico2):
    t = []
    for i in range(len(tinit)):
        if tinit[i][cledico2] == valeurdico2 :
            t.append(tinit[i])
    print(t)
    for i in range(1, len(t)):
        cle = t[i]
        k = i - 1
        while k >= 0 and cle[cledico1] < t[k][cledico1]:
            t[k + 1] = t[k]
            k -= 1
        t[k + 1] = cle
    return t
```

6. Expliquer les étapes de l'algorithme de tri à bulles sur l'exemple suivant en affichant pour chaque étape le contenu du tableau et en montrant explicitement quels éléments sont éventuellement modifiés. Explications succinctes attendues, utiliser des schémas (3 points)

```
t = tri_Bulles([8, 12, 3, 5, 21, 35])
```

7. Imaginons un fichier texte "`final.txt`" dans lequel sont inscrites les notes du final INF1 de votre groupe. Pour simplifier le problème, chaque ligne correspond à la note d'une personne mais le fichier ne contient aucune autre information (pas de nom, etc.). Le final ayant été trop difficile, proposer une fonction Python qui permettra de créer un deuxième fichier "`final_majore.txt`" dans lequel les notes seront majorées de 1 point à l'exception des notes 0 et 20. (2 points)

Après l'appel suivant : `ajout("final.txt", "final_majore.txt")`, les fichiers contiendront les notes suivantes :

"final.txt"	"final_majore.txt"
5	6
14	15
0	0
20	20
12	13