

**ECOLE SUPERIEURE D'INFORMATIQUE DE GESTION**



## Travail d'Algorithmes et Langage de Programmation Novembre 2024

### Informations générales

<b>Branche :</b>	<b>ALGORITHMES ET LANGAGE DE PROGRAMMATION (ALP)</b>
<b>Classes :</b>	<b>ESIG 1 Classes A et B (groupes 1, 2 et 3)</b>
<b>Date :</b>	<b>29 octobre 2024</b>

## Modalités

1. Durée : 180 minutes en tout (parties 1 et 2).
2. Travail individuel.
3. Documentation personnelle (livres, papiers et électronique) : Autorisée uniquement pour la partie 2 de l'épreuve (cf. ci-dessous).  
Aucune documentation n'est autorisée pour la partie 1.
4. La partie 1 dure 60 minutes au maximum et 30 minutes au moins. Aucune sortie aux toilettes n'est autorisée pendant la première heure. Tout le temps restant après la reddition de la partie 1 pourra être mis à profit pour la partie 2.
5. Tout partage de ressources de votre poste de travail avec le réseau ou toute autre tentative de communication (que ce soit avec un humain ou un outil d'intelligence artificielle) seront considérés comme de la fraude et sanctionnés par la note minimale. Cela comprend la présence d'un téléphone portable, d'objet connectés (montre, lunettes, ...), clé USB ou tout autre dispositif de stockage ou de communication à proximité immédiate de votre place de travail : tout ceci doit être posé à l'endroit indiqué par le/la surveillant-e.

## Démarrage et préparation

### **Avant le début de l'épreuve**

1. Connectez-vous au réseau sur le poste de travail qui vous a été attribué.
2. Copiez votre documentation personnelle (disque externe, clé USB, Google Drive, etc.) sur un emplacement local de la machine, tel que `C:\ESIGUsers\Doc`.
3. Copiez l'archive nommée `ALP-novembre-24.7z` depuis le répertoire `G:\ESII-Uldry\ESIG\Distribution\2024-2025\ESIG-1\ALP` vers le répertoire `C:\ESIGUsers`. Cette archive est protégée par un mot de passe qui vous sera communiqué après la coupure du réseau.
4. Eteignez l'écran en attendant de recevoir l'énoncé de la partie 1. L'écran pourra être rallumé pour la partie 2.

### **Une fois le mot de passe reçu et la partie 1 rendue**

1. Décompressez l'archive pour obtenir le répertoire `C:\ESIGUsers\ALP-novembre-24`
2. À l'intérieur du répertoire décompressé, renommez les fichiers `questionX.py` en y ajoutant vos nom et prénom de façon à avoir 3 fichiers `questionX_NomPrenom.py`.
3. Commencez la partie 2.

## Consignes générales

### **Sur l'organisation**

- Lisez tous les documents fournis.
- Complétez les différentes questions de manière à ce qu'elles répondent aux spécifications de l'énoncé.
- Rendez les fichiers correspondants sur la clé USB qui vous sera remise

### **Sur la programmation en Python**

- Interdiction d'utiliser les fonctions prédéfinies `max`, `min`, `index`, `count`, `sum`, `mean`, `filter` ou équivalent dans des modules fournis par python.
- Les éléments suivants sont explicitement autorisés : la fonction `len` et le mot clé `in` uniquement dans le contexte d'une boucle `for` (`for ... in ...`).
- Pas de résultats ou de données codés « en dur » : votre code doit pouvoir s'adapter à d'autres jeux de données.
- Les seules variables globales autorisées sont les constantes.

C'est à vous de vérifier que les fichiers enregistrés et rendus contiennent bien la dernière version de votre travail

## Partie 1 : Débranchée

Nom étudiant-e : .....

Prénom étudiant-e : .....

Professeur : .....



Si vous détachez les feuilles de l'énoncé Partie 1 :  
écrivez **sur chaque page** vos **nom et prénom**.

**Aucune** documentation n'est autorisée pour cette partie

*L'écran de votre ordinateur doit être **éteint***

## Question – Lire du code

### 1 Analyse de bouts de codes

#### 1.1 Lequel des codes ci-dessous pourrait être placé à la ligne No 3?

```
1 nb = input("Veuillez entrer un nombre:");  
2 if int(nb) % 2 == 0:  
3     # Laquelle des propositions peut être placée ici?
```

Veuillez cocher une seule réponse :

- ☐ `print("Le nombre entré est un nombre entier");`
- ☐ `print("Le nombre entré est un nombre pair");`
- ☐ `print("Le nombre entré est un nombre à virgule");`

#### 1.2 Qu'affiche la console dans l'extrait de code ci-dessous ?

```
var1 = 3  
var2 = 8  
var3 = var1 * var2 - 8  
print(var3)  
  
var4 = var3 - var2 / 4  
print(var4)  
  
res = var3 + var4  
print(res)
```

Output de la console :

#### 1.3 Combien valent les variables *res* et *cpt* à la fin de l'algorithme ci-dessous ?

```
def mes_boucles(c, r):  
    while c < 3:  
        while r % 3 > 0:  
            r += 1  
        c += 1  
    return r, c  
  
cpt = 1  
res = 1  
cpt, res = mes_boucles(cpt, res)
```

Valeur de *res* :

Valeur de *cpt* :

## 2 Analyse de fonctions et procédures

### 2.1 Analysez le code ci-dessous et écrivez l'output de la console python

```
def f_a(n):  
    print("n:", n)  
    n = n*3  
    f_b(6, n)  
  
def f_b(a,b):  
    a = b / 2  
    b = a - 2  
    print("a:", a)  
    print("b:", b)  
  
def main():  
    print("----")  
    f_a(8)  
    print("----")  
  
if __name__ == "__main__":  
    main()
```

Output de la console :

### 2.2 Analysez le code ci-dessous et écrivez l'output de la console python

```
def f_a(n):  
    print("n:", n)  
    resb = f_b(10, n*2)  
    resc = f_c(n, resb)  
    return resc  
  
def f_b(a,b):  
    print("a:", a, "b:", b)  
    res = (a + b) / 2  
    print("res:", res)  
    return res  
    print("fin f_b")  
  
def f_c(n, val):  
    print("n:", n, "val:", val)  
    res_c = val - n  
    print("res_c", res_c)  
    return res_c  
  
def main():  
    print("----")  
    nb_secret = 21  
    resultat = f_a(nb_secret)  
    print("resultat: ", resultat)  
  
if __name__ == "__main__":  
    main()
```

Output de la console :

### 3 Complétion et correction de code

3.1 Le code ci-dessous provoque une erreur à l'exécution. Vous avez la trace de l'erreur. Spécifiez la/les numéros de ligne à corriger et proposez une correction.

```
1 def main():
2     print("----")
3     print(a)
4     a = 9
5     print("----")
6
7 if __name__ == "__main__":
8     main()
```

Shell

```
>>> %Run -c $EDITOR_CONTENT
----
Traceback (most recent call last):
  File "<string>", line 8, in <module>
  File "<string>", line 3, in main
UnboundLocalError: local variable 'a' referenced before assignment
>>>
```

Votre réponse

**3.2 Le code ci-dessous provoque une erreur à l'exécution. Vous avez la trace de l'erreur. Spécifiez la/les numéros de ligne à corriger et proposez une correction.**

```
1 def calcul_prix(prix_depart):
2     prix_final = prix_depart + prix_depart * 10/100
3
4 def main():
5     print("----")
6     a = 18
7     calcul_prix(a)
8     print(prix_final)
9
10 if __name__ == "__main__":
11     main()
```

```
hell -
>> %Run -c $EDITOR_CONTENT
-----
Traceback (most recent call last):
  File "<string>", line 12, in <module>
  File "<string>", line 8, in main
NameError: name 'prix_final' is not defined
>>
```

Votre réponse

**3.3 Le code ci-dessous n'affiche rien du tout. On souhaite qu'il affiche les valeurs suivantes : 10, 8, 6, 4, 2. Spécifiez la/les numéros de ligne à corriger et proposez une correction qui permet d'afficher le résultat attendu**

```
1 def main():  
2     for i in range(0, 10, -2):  
3         print(i)  
4  
5 if __name__ == "__main__":  
6     main()
```

Votre réponse



## Partie 2 : Programmation Python

**Nom étudiant-e** : .....

**Prénom étudiant-e** : .....

**Professeur** : .....

**Numéro du poste** : .....

**Numéro de la clé USB** : .....



La documentation personnelle est autorisée pour cette partie.

N'oubliez pas de sauvegarder régulièrement votre travail.

N'oubliez pas de renommer les fichiers questionX.py en y ajoutant vos nom et prénom à la fin.

## Question 1 : Système de tarification d'une bibliothèque

Votre travail s'effectuera dans le fichier `question1_NomPrenom.py`

Vous travaillez pour une bibliothèque municipale qui souhaite mettre en place un nouveau système de tarification pour l'emprunt de livres. On vous demande de créer un programme qui calcule le coût total pour un emprunt de livres. Ce coût dépend de la catégorie du livre emprunté, du nombre de jours d'emprunt, et du type d'adhérent.

La bibliothèque propose trois catégories de livres (numérotées 1, 2 et 3) :

1. Livres grand public
2. Nouveautés
3. Ouvrages rares ou spécialisés

Il existe deux types d'adhérents :

- Adhérents standard
- Adhérents premium (qui bénéficient d'avantages)

Les règles de calcul du coût total d'emprunt sont les suivantes :

- Chaque emprunt comporte des frais de gestion forfaitaires : 1 CHF pour les livres de catégorie 1, 1.50 CHF pour la catégorie 2, et 3 CHF pour la catégorie 3.
- Le tarif journalier dépend de la catégorie du livre : 0.10 CHF pour la catégorie 1, 0.20 CHF pour la catégorie 2, et 0.30 CHF pour la catégorie 3.
- Les adhérents premium bénéficient d'une réduction de 20% sur le coût total (frais de gestion + tarif journalier).
- Le coût total d'un emprunt ne peut pas dépasser 24 CHF, quel que soit le type d'adhérent ou la catégorie de livre.

Votre programme doit :

- Demander à l'utilisateur de saisir la catégorie du livre (1, 2 ou 3), le type d'adhérent (standard, premium), et le nombre de jours d'emprunt.
- Valider les données saisies et afficher des messages d'erreur si nécessaire (voir résultats attendus).
- Calculer le coût total de l'emprunt selon les règles énoncées.
- Afficher un récapitulatif comprenant la catégorie du livre, le type d'adhérent, le nombre de jours d'emprunt, et le coût total à payer. Si le coût a dépassé le maximum de 24 CHF un message doit indiquer qu'il a été plafonné (voir exemple de sorties).

Votre programme doit être structuré en fonctions et procédures pour faciliter la lecture et la maintenance du code.

Un minimum de 3 fonctions ou procédures est requis en plus de la procédure `main` et d'une éventuelle fonction pour lire les saisies de l'utilisateur `donnees` ou similaire.

## Résultats attendus :

Voici quelques exemples de résultats attendus. Assurez-vous de tester votre programme avec ces valeurs et d'autres cas de figure.

```
Entrez la catégorie du livre (1, 2 ou 3): 4
Entrez le type d'adhérent (standard ou premium): 2
Entrez le nombre de jours d'emprunt (max. 92): 93
Erreur: catégorie de livre invalide (doit être 1, 2 ou 3)
Erreur: type d'adhérent invalide (doit être standard ou premium)
Erreur: nombre de jours invalide (doit être inférieur à 93)
```

```
Entrez la catégorie du livre (1, 2 ou 3): 1
Entrez le type d'adhérent (standard ou premium): standard
Entrez le nombre de jours d'emprunt (max. 92): 0
Erreur: nombre de jours invalide (doit être supérieur à 0)
```

```
Entrez la catégorie du livre (1, 2 ou 3): 1
Entrez le type d'adhérent (standard ou premium): standard
Entrez le nombre de jours d'emprunt (max. 92): 70
Catégorie: Livres grand public
Type d'adhérent: standard
Nombre de jours: 70
Coût total: 8.0 CHF
```

```
Entrez la catégorie du livre (1, 2 ou 3): 2
Entrez le type d'adhérent (standard ou premium): premium
Entrez le nombre de jours d'emprunt (max. 92): 30
Catégorie: Nouveautés
Type d'adhérent: premium
Nombre de jours: 30
Coût total: 6.0 CHF
```

```
Entrez la catégorie du livre (1, 2 ou 3): 3
Entrez le type d'adhérent (standard ou premium): standard
Entrez le nombre de jours d'emprunt (max. 92): 71
Catégorie: Ouvrages rares ou spécialisés
Type d'adhérent: standard
Nombre de jours: 71
Coût total: 24.3 CHF (plafonné à 24.0 CHF)
```

## Question 2 : Calcul des pénalités de retard

Votre travail s'effectuera dans le fichier `question2_NomPrenom.py`

La bibliothèque municipale met en place un nouveau système de pénalités pour les retours tardifs de livres. On vous demande de créer un programme qui calcule le montant total des pénalités pour un adhérent qui a rendu des livres en retard.

Les règles sont les suivantes :

- La pénalité de base est de 0.50 CHF par livre et par jour de retard.
- Pour chaque semaine complète de retard, on ajoute 10% au total de la pénalité du livre.
- La pénalité est calculée sur une base de 7 jours par semaine.

Votre programme doit :

- Demander le nombre de jours de retard.
- Demander le nombre de livres rendus en retard.
- Utiliser une boucle pour itérer sur le nombre de jours de retard :
  - Ajouter la pénalité de ce jour au total général
  - Si le jour marque la fin d'une semaine complète, ajouter 10% à la pénalité accumulée.
  - Afficher le résultat intermédiaire
- Afficher le montant total des pénalités à payer, arrondi aux 5 centimes près.

Aide :

Utilisez la fonction `round` pour arrondir le montant final aux 5 centimes près, comme suit :

```
penalite = round(penalite * 20) / 20.0
```

**Résultats attendus :**

Entrez le nombre de jours de retard: 10

Entrez le nombre de livres rendus en retard: 2

Calcul des pénalités

Jour 1: 1.0 CHF

Jour 2: 2.0 CHF

Jour 3: 3.0 CHF

Jour 4: 4.0 CHF

Jour 5: 5.0 CHF

Jour 6: 6.0 CHF

Jour 7: 7.0 CHF + 10% (0.700000000000000001) = 7.7 CHF

Jour 8: 8.7 CHF

Jour 9: 9.7 CHF

Jour 10: 10.7 CHF

Montant total des pénalités à payer: 10.7 CHF

Entrez le nombre de jours de retard: 20

Entrez le nombre de livres rendus en retard: 3

Calcul des pénalités

Jour 1: 1.5 CHF

Jour 2: 3.0 CHF

Jour 3: 4.5 CHF

Jour 4: 6.0 CHF

Jour 5: 7.5 CHF

Jour 6: 9.0 CHF

Jour 7: 10.5 CHF + 10% (1.05) = 11.55 CHF

Jour 8: 13.05 CHF

Jour 9: 14.55 CHF

Jour 10: 16.05 CHF

Jour 11: 17.55 CHF

Jour 12: 19.05 CHF

Jour 13: 20.55 CHF

Jour 14: 22.05 CHF + 10% (2.205) = 24.2550000000000003 CHF

Jour 15: 25.7550000000000003 CHF

Jour 16: 27.2550000000000003 CHF

Jour 17: 28.7550000000000003 CHF

Jour 18: 30.2550000000000003 CHF

Jour 19: 31.7550000000000003 CHF

Jour 20: 33.255 CHF

Montant total des pénalités à payer: 33.25 CHF

## Question 3

Votre travail s'effectuera dans le fichier `question3_NomPrenom.py`

Vous n'avez pas le droit de modifier le code existant.

Vous complétez le script en remplaçant les instructions `pass` des 2 procédures à coder par du code fonctionnel respectant les spécifications données.

### A. Affichage des premières durées de prêt valides

Dans la bibliothèque municipale, vous avez une liste contenant les durées de prêt (en jours) des livres. Certaines durées peuvent être égales à 0 (ce qui signifie que le livre n'a pas encore été emprunté ou qu'il est en attente). Vous devez afficher les `nb` premières durées non nulles, puis indiquer combien de durées ont été affichées et combien ont été examinées au total.

#### Spécifications :

Complétez la procédure `afficher_premieres_durees(lst_durees, nb)` qui :

- Prend en paramètre une liste des durées de prêt (`lst_durees`) et un nombre (`nb`) représentant combien de durées valides doivent être affichées.
- Parcourt la liste des durées et affiche les `nb` premières valeurs non nulles (durées différentes de 0).
- Affiche également le nombre total de valeurs affichées et le nombre total de valeurs examinées.

### B. Calcul du total des durées de prêt

Dans la bibliothèque municipale, vous avez une liste des livres empruntés, une liste des livres existants et une liste des durées de prêt associées à chaque livre. Un utilisateur peut emprunter plusieurs livres, et votre programme doit calculer la durée totale des prêts des livres empruntés.

Le programme doit parcourir la liste des livres empruntés, rechercher chaque livre dans la liste des livres existants, et additionner les durées de prêt correspondantes.

#### Spécifications :

Complétez la procédure `calculer_duree_totale(lst_emprunts, lst_livres, lst_durees)` qui :

- Prend en paramètre trois listes :
  - `lst_emprunts` : La liste des livres empruntés.
  - `lst_livres` : La liste des livres existants, c'est-à-dire, la liste de l'ensemble des livres dans l'inventaire de la bibliothèque (empruntés et non empruntés).
  - `lst_durees` : La liste des durées de prêt associées à chaque livre (dans le même ordre que `lst_livres`).
- Pour chaque livre emprunté, si le livre est présent dans la liste des livres existants, ajoute sa durée de prêt à un total.
- Si un livre emprunté n'est pas trouvé dans la liste des livres existants, affiche un message signalant que ce livre est inconnu.
- Affiche la durée totale des prêts des livres empruntés qui sont trouvés dans la liste des livres existants.

## Résultats attendus

Exemple de sorties produites, si on souhaite n'afficher que **2** valeurs :

Entrez un nombre de durées non-nulles à afficher (Partie A)2  
Liste des durées de prêt : [30, 0, 0, 80, 20, 7]

Partie A

Durées de prêt : 30 80 (2 valeurs affichées sur 4 examinées)

Partie B

Le livre "Harry Potter" est inconnu dans la liste des livres!

Le livre "Un palais d'épines et de roses" est inconnu dans la liste des livres!

La durée totale des prêts des livres empruntés est : 50 jours.

Exemple de sorties produites, si on souhaite n'afficher que **4** valeurs :

Entrez un nombre de durées non-nulles à afficher (Partie A)4  
Liste des durées de prêt : [30, 0, 0, 80, 20, 7]

Partie A

Durées de prêt : 30 80 20 7 (4 valeurs affichées sur 6 examinées)

Partie B

Le livre "Harry Potter" est inconnu dans la liste des livres!

Le livre "Un palais d'épines et de roses" est inconnu dans la liste des livres!

La durée totale des prêts des livres empruntés est : 50 jours.

Exemple de sorties produites, si on souhaite n'afficher que **20** valeurs :

Entrez un nombre de durées non-nulles à afficher (Partie A)20  
Liste des durées de prêt : [30, 0, 0, 80, 20, 7]

Partie A

Durées de prêt : 30 80 20 7 (4 valeurs affichées sur 6 examinées)

Partie B

Le livre "Harry Potter" est inconnu dans la liste des livres!

Le livre "Un palais d'épines et de roses" est inconnu dans la liste des livres!

La durée totale des prêts des livres empruntés est : 50 jours.