

**CONCOURS EXTERNE
DE TECHNICIEN PRINCIPAL
DE POLICE TECHNIQUE ET SCIENTIFIQUE
DE LA POLICE NATIONALE**

SESSION 2025

INFORMATIQUE DÉVELOPPEMENT LOGICIEL

Épreuve écrite de connaissances
se rapportant à la spécialité choisie

Durée de l'épreuve : 3 heures - Coefficient : 2 – Note inférieure à 5/20 éliminatoire

AVERTISSEMENT :

Vous devez vous conformer aux consignes qui vous ont été données par le responsable de salle pour cette épreuve.

1. Vous devez obligatoirement et uniquement utiliser **un stylo bille à encre foncée (NOIRE ou BLEUE)**, et conserver la même couleur durant toute l'épreuve.

Il est strictement interdit d'utiliser dans votre copie :

- tout liquide correcteur ou effaceur ;
- un stylo à friction ;
- un stylo d'une autre couleur (rouge, vert, etc...) ;
- un stylo plume ;
- un crayon à papier ;
- un surligneur.

2. Aucun matériel n'est autorisé
3. Cette épreuve comporte 4 exercices auxquels il vous appartient de répondre avec clarté sur la copie qui vous a été distribuée en même temps que ce sujet.
4. Il vous appartient de vous assurer que le sujet en votre possession comporte la totalité des pages (4 pages).

Le sujet est noté sur 20 points

1 point évaluera la qualité rédactionnelle, la présentation et l'orthographe

Vous ne devez faire apparaître aucun signe distinctif dans votre copie, **ni votre nom ou un nom fictif, ni une signature ou un paraphe, ni le nom d'un commissariat ou ville, ni de sigles, ni de modèles de véhicule ainsi que des immatriculations, à l'exception de ceux contenus dans le sujet.**

**LE NON-RESPECT DE CES RÈGLES EST SUSCEPTIBLE D'ENTRAÎNER
L'ANNULATION DE LA COPIE PAR LE JURY NATIONAL.**

Exercice 1 (4,5 points)

Le laboratoire de criminalistique gère les correspondances entre des empreintes digitales relevées sur des scènes de crime et des suspects enregistrés.

Schéma relationnel :

SUSPECTS(idSuspect, nom, prenom, dateNaissance)

EMPREINTES(idEmpreinte, lieu, dateRelevee)

CORRESPONDANCES(#idEmpreinte, #idSuspect, tauxCorrespondance)

Travail demandé

Écrire les requêtes SQL pour répondre aux questions suivantes.

1. Lister les suspects ayant plus de trois correspondances avec un taux > 80 %.
2. Afficher toutes les empreintes prélevées dans des lieux pour lesquels il existe au moins deux dates de relevé différentes.
3. Calculer le taux moyen de correspondance pour chaque suspect.
4. Supprimer les correspondances dont le taux est inférieur à 30 %.
5. Afficher les suspects n'ayant aucune correspondance enregistrée.

Exercice 2 (5,5 points)

Pour cet exercice vous êtes libre de choisir le langage de programmation que vous maîtrisez. On souhaite analyser des chaînes ADN en recherchant des séquences spécifiques dans un échantillon.

L'objectif est de concevoir une fonction qui identifie toutes les occurrences d'un ou plusieurs motifs dans une séquence ADN donnée.

Travail demandé

Écrire une fonction nommée *rechercher_sequences* qui :

- prend en paramètre une chaîne de caractères représentant une séquence ADN (exemple : "ATCGATCGA") et une liste de motifs à rechercher dans cette séquence (exemple : ["ATC", "CGA"]),
- retourne une structure de données (de type dictionnaire, tableau associatif ou équivalent selon le langage choisi) dans laquelle chaque motif est associé à la liste des positions où il apparaît dans la chaîne.

Exemple :

```
>>> rechercher_sequences("ATCGATCGA", ["ATC", "CGA"])
{"ATC": [0, 4], "CGA": [2, 6]}
```

1. Implémenter la fonction *rechercher_sequences* selon les spécifications ci-dessus.

2. Adapter votre fonction pour qu'elle recherche également les motifs complémentaires de l'ADN.

Dans l'ADN, chaque base a un complément :

A ↔ T

T ↔ A

C ↔ G

G ↔ C

Par exemple, le complément de "ATC" est "TAG".

La fonction devra retourner, pour chaque motif, deux listes de positions : l'une pour le motif original, et l'autre pour son complément.

Exemple :

```
>>> rechercher_sequences("ATCGATCGATAG", ["ATC"])
{"ATC": { "motif": [0, 4], "complement": [9] } }
```

3. Proposer un test unitaire sur la nouvelle fonction *rechercher_sequences*.

Exercice 3 (4 points)

1. Qu'est-ce que la CNIL ?
2. Quelle est la différence entre un commit et un push dans Git ?
3. Expliquez le rôle d'un ORM dans une application web.
4. Qu'est-ce qu'une authentification forte ?

Exercice 4 (5 points)

Pour cet exercice vous êtes libre de choisir le langage de programmation que vous maîtrisez.

On souhaite modéliser un système de notifications. Chaque notification doit pouvoir être envoyée et affichée.

1. Créer une classe abstraite Notification avec une méthode envoyer() à surcharger.
2. Implémenter les classes EmailNotification, SMSNotification, PushNotification. Chacune doit surcharger envoyer() et ajouter une méthode afficher().
3. Ajouter à chaque classe une priorité (entier entre 1 et 3) accessible via une méthode get_priorite().
4. Écrire une fonction `traiter_notifications(liste_notifs)` qui trie les notifications selon la priorité et appelle envoyer() et afficher() pour chacune.
5. Implémenter la classe NotificationUrgente, qui hérite de la classe EmailNotification.

Cette notification urgente doit respecter les règles suivantes :

- Sa priorité est toujours fixée à 0, même si une autre valeur est transmise lors de l'instanciation.
- Lors de l'envoi et de l'affichage, un préfixe [URGENT] doit être ajouté au message standard.