



**CONCOURS INTERNE**  
**POUR LE RECRUTEMENT DE CONTRÔLEURS DES DOUANES ET DROITS INDIRECTS**  
  
**BRANCHE DU CONTRÔLE DES OPÉRATIONS COMMERCIALES ET D'ADMINISTRATION GÉNÉRALE**  
  
**DES 24 ET 25 FÉVRIER 2014**

**ÉPREUVE ÉCRITE D'ADMISSIBILITÉ N°2**

(DURÉE : 5 HEURES - COEFFICIENT 5)

**OPTION D : INFORMATIQUE**

**PUPITREUR : page 2**

**PROGRAMMEUR : page 3**

**AVERTISSEMENTS IMPORTANTS**

Si, lors de votre inscription, vous avez **choisi d'obtenir la qualification de pupitreur**, vous devez impérativement composer dans le seul sujet permettant d'obtenir cette qualification sinon votre copie sera notée zéro.

Si, lors de votre inscription, vous avez **choisi d'obtenir la qualification de programmeur**, vous devez impérativement composer dans le seul sujet permettant d'obtenir cette qualification sinon votre copie sera notée zéro.

Veillez à bien indiquer sur votre copie **la qualification** pour laquelle vous allez composer ainsi que le **nombre d'intercalaires** utilisé (la copie double ne compte pas).

**A l'exception de l'organigraphe**, l'usage de tout matériel autre que celui d'écriture et de tout document autre que le support fourni est interdit. Toute fraude ou tentative de fraude constatée par la commission de surveillance entraînera **l'exclusion du concours**.

Il vous est interdit de quitter définitivement la salle d'examen **avant le terme de la première heure**.

Le présent document comporte 5 pages numérotées.

## **SUJET POUR OBTENIR LA QUALIFICATION DE PUPITREUR**

### **Remarques préliminaires :**

- ***Toutes les questions doivent être traitées. Chaque réponse devra être précédée du numéro de la question à laquelle elle se rapporte.***
- ***Toutes les réponses doivent être justifiées.***
- ***Tous les schémas et diagrammes éventuellement réalisés doivent être accompagnés d'un commentaire expliquant la valeur de leurs symboles.***

### **Question n° 1 :**

Quels contrôles l'employeur peut-il faire de l'utilisation des ressources informatiques par ses salariés ?

### **Question n° 2 :**

Quels sont les principes de fonctionnement du modèle OSI ?  
(OSI = Open Systems Interconnection - interconnexion des systèmes ouverts)

### **Question n° 3 :**

Présentez les principales topologies de réseau.

### **Question n° 4 :**

À quoi correspond l'adresse MAC d'un matériel ?  
(MAC = Media Access Control – contrôle d'accès au support)

### **Question n° 5 :**

Quelle est la différence entre un routeur et une passerelle ?

---

## **SUJET POUR OBTENIR LA QUALIFICATION DE PROGRAMMEUR**

### **Remarques préliminaires :**

- ***Vous devez traiter le sujet dans le langage choisi lors de votre inscription. Si vous traitez le sujet dans un langage différent, votre copie sera notée zéro.***
- ***Vous préciserez au début de votre devoir le langage choisi.***
- ***Toutes les questions doivent être traitées. Chaque réponse devra être précédée du numéro de la question à laquelle elle se rapporte.***
- ***Toutes les réponses doivent être justifiées.***
- ***Si des options vous semblent nécessaires après la lecture de tous les éléments fournis, il vous appartiendra de les indiquer et de les justifier.***
- ***Tous les schémas et diagrammes à réaliser doivent être accompagnés d'un commentaire expliquant la valeur de leurs symboles.***

Soit un drone permettant de lire sur la route les plaques d'immatriculation des véhicules routiers et de les comparer avec une liste de plaques d'immatriculation de véhicules volés. Si une plaque d'immatriculation lue correspond à celle d'un véhicule volé, le drone indique à l'opérateur le numéro de la plaque d'immatriculation ainsi que sa localisation GPS.

Actuellement, cette liste est enregistrée dans un fichier ASCII où les numéros sont séparés par un retour chariot (voir **Figure 1**). Le drone bride la taille du fichier contenant la liste des numéros de plaques d'immatriculation à 2 kilo-octets.

Par souci de simplicité, on considère que toutes les plaques d'immatriculation sont au format suivant : 12-NNN-34 (deux chiffres, un tiret, trois lettres majuscules, un tiret et deux chiffres).

### ***Figure 1 :***

15-SOE-86  
97-PEN-13  
42-NEO-42

Le drone donne la possibilité de modifier les deux sous-programmes suivants :

`ChargerFichier` : qui prend en argument le nom du fichier contenant la liste des numéros de plaques d'immatriculation des véhicules volés et qui retourne un pointeur sur une structure de données. Le travail de ce sous-programme est de lire le fichier et de charger l'ensemble des numéros dans une structure de données non définie par le constructeur du drone.

`FiltererPlaques` : qui prend en argument un pointeur sur la structure retournée par `ChargerFichier` ainsi qu'une liste chaînée contenant les numéros récemment lus par les caméras du drone. Son travail est de retourner une nouvelle liste chaînée contenant l'ensemble des numéros contenus à la fois dans la structure de données retournée par `ChargerFichier` et dans la liste chaînée passée en argument.

Les listes chaînées de numéros de plaques d'immatriculation manipulées par l'API du drone sont de type `ListeNumeros` qui sont elles-mêmes composées de nœuds de type `NoeudNumero`. Le détail de ces deux structures est donné dans la **Figure 2**. La notation `->T data` signifie que la donnée `data` est un pointeur sur une structure de type `T`.

**Figure 2 :**

```
Structure ListeNumeros
->NoeudNumero premierElement
```

```
Structure NoeudNumero
ChaîneCaractère numero
->NoeudNumero suivant
```

Actuellement, chaque numéro de plaque d'immatriculation enregistré dans le fichier représente 10 octets, retour chariot compris. Il est possible que chaque numéro n'utilise plus que 5 octets en les enregistrant sous format binaire. En utilisant une représentation optimisée, il est même possible de descendre en dessous de 5 octets par numéro.

L'objectif de l'exercice est d'écrire les algorithmes ainsi que les sous-programmes associés permettant :

De prendre en argument un fichier texte contenant une liste de numéros de plaques d'immatriculation de véhicules volés et de le transcoder en un fichier binaire de moins de 2 kilo-octets. Ce sous-programme s'appellera `TranscoderFichier`.

De réécrire le sous-programme `ChargerFichier` afin qu'il prenne en argument le nom du fichier précédemment transcodé avec `TranscoderFichier`.

De réécrire le sous-programme `FiltererPlaques` afin qu'il utilise la même structure de données que celle retournée par votre implémentation de `ChargerFichier`.

D'un point de vue optimisation d'exécution des sous-programmes, on considérera que la vitesse d'exécution de `FiltererPlaques` est prioritaire sur celle de `ChargerFichier` car ce dernier n'est exécuté que très rarement.

**Travail demandé :**

1) Décrivez en 10 lignes maximum la stratégie que vous utiliserez pour enregistrer la liste des numéros de plaques d'immatriculation en un minimum de bits. Puis décrivez la structure du fichier binaire que vous utiliserez. Les algorithmes de compactage n'étant pas au programme de l'exercice, vous devrez considérer que tous les chiffres sont équiprobables. Il en sera de même pour les lettres.

2) Écrivez l'algorithme du sous-programme `TranscoderFichier`. Ce sous-programme a deux arguments qui correspondent au nom du fichier texte qui servira d'entrée et au nom du fichier binaire qui contiendra le résultat du transcodage.

3) Traduisez l'algorithme du sous-programme `TranscoderFichier` dans le langage de programmation choisi.

4) Indiquez la structure de données (tableau, arbre ...) qui servira à transférer la liste des numéros entre les sous-programmes `ChargerFichier` et `FiltererPlaques`.

Préciser en 5 lignes maximum en quoi la structure de données choisie est optimum dans notre cas. Il est rappelé que les performances de `FiltererPlaques` sont prioritaires à celles de `ChargerFichier`. Il est supposé que le drone a suffisamment de mémoire pour pouvoir allouer 128 mégaoctets à la structure de données.

5) Expliquez le fonctionnement de la structure de données choisie. Décrivez dans un langage algorithmique semblable à celui de la *figure 2* les différentes structures la composant.

6) Écrivez les deux algorithmes permettant respectivement d'ajouter un numéro de plaque à votre structure de données et de chercher l'existence d'un numéro dans la structure de données.

7) Codez les deux algorithmes du 6) ainsi que les structures du 5) en langage de programmation. Il vous est demandé de coder vous-même ces algorithmes et non d'utiliser des structures de données fournies par le langage de programmation ou son API.

8) Écrivez l'algorithme du sous-programme `ChargerFichier`.

9) Codez l'algorithme du sous-programme `ChargerFichier`.

10) Écrivez l'algorithme du sous-programme `FiltererPlaques`.

11) Après avoir codé les structures de données `ListeNumeros` et `NoeudNumero` de la Figure 2 dans le langage de programmation, codez le sous-programme `FiltererPlaques`.

---