

M10/5/COMSC/SP2/FRE/TZ0/XX



22107018



International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

**INFORMATIQUE
NIVEAU MOYEN
ÉPREUVE 2**

Vendredi 7 mai 2010 (matin)

1 heure 30 minutes

INSTRUCTIONS DESTINÉES AUX CANDIDATS

- N'ouvrez pas cette épreuve avant d'y avoir été autorisé(e).
- Répondez à toutes les questions.

Répondez à *toutes* les questions.

1. (a) Suggérez une structure de données adaptée au stockage des réponses à un ensemble de 100 questions et dans laquelle chaque réponse est soit « oui » soit « non ». Donnez des raisons justifiant votre suggestion.

[4 points]

- (b) À l'aide de la structure de données suggérée dans la partie (a), construisez la méthode `totalOui()` qui renverra le nombre total de réponses « oui ». La méthode a été initiée ci-dessous.

```
public int totalOui()
{
    // lignes de code manquantes
}
```

[4 points]

La méthode `triNombre()` reçoit un tableau d'`int` de longueur inconnue (mais contenant 100 éléments maximum), et place les entiers dans trois tableaux différents `a`, `b` et `c`, où

`a` = entiers négatifs
`b` = entiers pairs non-négatifs
`c` = entiers qui ne sont ni dans `a` ni dans `b`.

La méthode renvoie alors le nombre d'éléments du tableau `b`.

- (c) Construisez la méthode `triNombre()`, qui a été initiée ci-dessous. Vous pouvez supposer que les tableaux sont initialement vides.

```
public int triNombre(int[] original)
{
    // lignes de code manquantes
    if (...)
        // lignes de code manquantes
    else
        // lignes de code manquantes
}
```

[8 points]

- (d) Sans écrire de code, suggérez comment la méthode `triNombre()` pourrait être modifiée afin qu'elle renvoie non seulement le nombre total d'éléments du tableau `b`, mais également le nombre total d'éléments du tableau `a`. Aucune méthode supplémentaire ne doit être envisagée.

[4 points]

2. Un *fichier maître* a été créé qui contient les informations relatives aux joueurs d'une ligue de football. Chaque enregistrement du fichier contient le nom du joueur et le nombre de buts marqués. Les enregistrements sont stockés par ordre alphabétique des noms des joueurs.

Les trois premiers enregistrements sont donnés ci-dessous :

Enregistrement 0	
nom	buts
Adams P.	5

Enregistrement 1	
nom	buts
Alvez J.	1

Enregistrement 2	
nom	buts
Aziz M.	3

Le fichier maître est mis à jour chaque semaine par un *fichier de détail*.

- (a) Exposez les différences entre un fichier maître et un fichier de détail. [2 points]

Chaque semaine, le processus suivant a lieu :

- un utilisateur entre des données dans un nouveau fichier de détail dont la structure d'enregistrement est identique à celle du fichier maître et qui contient les informations sur les buteurs de cette semaine
- le fichier de détail est trié dans le même ordre que le fichier maître
- le fichier maître est mis à jour par le fichier de détail
- le nombre total de buts marqués cette semaine est imprimé.

- (b) Dessinez un *ordinogramme de système* représentant le processus ci-dessus. [6 points]

Pour que le processus de mise à jour ait lieu, les enregistrements du fichier maître sont lus dans le tableau `maître[]`, qui est un tableau d'objets `Joueur`. Un objet `Joueur` est défini par la classe `Joueur`, montrée ci-dessous.

```
class Joueur
{
    public String nom;    // nom du joueur
    public int but;      // nombre de buts marqués par ce joueur
}
```

Les enregistrements de ce fichier de détail sont lus dans le tableau `détail[]` qui est également un tableau d'objets `Joueur`.

- (c) Expliquez pourquoi il est important que le fichier maître et le fichier de détail soient dans le même ordre. [2 points]

(Suite de la question à la page suivante)

(Suite de la question 2)

(d) En faisant référence aux données indiquées dans le fichier maître original, indiquez la valeur de

(i) `maitre[0].nom` ; *[1 point]*

(ii) `maitre[2].buts`. *[1 point]*

Le processus de mise à jour est comme suit :

répéter pour chaque joueur dans le tableau `detail[]`

- parcourir séquentiellement le table `maitre[]` jusqu'à ce que le même joueur soit trouvé
- mettre à jour le nombre de buts marqués par ce joueur.

La méthode `miseajour()` réalise ce processus et renvoie **également** le nombre total de buts marqués cette semaine.

(e) Construisez la méthode `miseajour()`, qui a été initiée ci-dessous.

```
// les deux tableaux sont déclarés en tant que variables de classe (globales)
public int miseajour()
{
    // lignes de code manquantes
}
```

Vous pouvez supposer que tous les joueurs qui apparaissent dans le fichier de détail se trouvent déjà dans le fichier maître.

[8 points]

3. Cette question nécessite la lecture de l'étude de cas.

- (a) La construction du Terminal 5 de l'aéroport de Heathrow (Londres) incluait une nouvelle tour de contrôle pour tout l'aéroport. La bascule entre les deux tours a été réalisée en exécutant le nouveau système parallèlement à l'ancien.
- (i) Exposez ce qu'implique l'exécution de systèmes en parallèle. [2 points]
- (ii) Exposez **une** raison justifiant le choix d'une forme parallèle de bascule et pas d'une autre. [2 points]
- (b) Suggérez pourquoi les autorités de l'aéroport ont choisi la stratégie de test exposée à la page 4 de l'étude de cas. [2 points]
- (c) Décrivez **deux** procédures de sécurité différentes qui contribueraient à empêcher l'accès physique de personnes non autorisées dans les sections à accès restreint de l'aéroport. [4 points]
- (d) (i) Identifiez un système d'aéroport qui utiliserait un traitement en temps réel. [1 point]
- (ii) Expliquez pourquoi le système identifié dans la partie (i) nécessiterait un traitement en temps réel. [3 points]
- (e) Un système informatique est conçu pour remplacer les strips en papier utilisés par le contrôle de la circulation aérienne. L'un des soucis des contrôleurs aériens était que l'interaction avec un système informatique pourrait les distraire de leurs tâches.
- (i) Expliquez pourquoi la *reconnaissance vocale* **ne** conviendrait **pas** comme *interface-utilisateur*. [3 points]
- (ii) Décrivez une interface-utilisateur adaptée. [3 points]
- (iii) Décrivez **une** caractéristique utile qui pourrait être incorporée dans la nouvelle interface utilisateur, et qui n'était pas présente dans l'interface originale. [2 points]
- (iv) Suggérez **une** stratégie de sauvegarde possible en cas de panne du système dans le contrôle de la circulation aérienne. [2 points]
- (f) En étudiant les deux points de vue de la discussion, examinez si les gouvernements doivent ou non avoir accès aux fichiers des « dossiers passagers (DP) » (voir page 10 de l'étude de cas). [6 points]