

Session 2013

## Examen OC INFORMATIQUE

---

### Informations et consignes :

- Temps à disposition : 3 heures.
  - Formulaire annexé autorisé.
  - Calculatrice non programmable autorisée.
  - Le nombre de points est indiqué pour chaque problème. Il y a 80 points au total.
  - 80 points donnent la note maximale (6).
  - Vous devez écrire proprement au stylo ou à l'encre.
  - Utilisez un feuillet par problème.
  - Ecrivez votre nom sur chaque feuille.
  - Rendez tous vos documents, y compris la donnée et le formulaire.
- 

### Exercice 1 : Logique (12 points)

Soit l'ensemble des codes binaires à 4 bits. Créer un circuit logique qui détecte les codes contenant plusieurs « 0 » consécutifs.

1. Établir la table de vérité.
2. En tirer la fonction logique correspondante et la simplifier.
3. Esquisser le circuit correspondant.

### Exercice 2 : Base de données (22 points)

Une fédération sportive souhaite informatiser son championnat. Pour cela, elle donne les consignes suivantes à l'informaticien qui se chargera du travail :

- La fédération est constituée de plusieurs clubs qui possèdent chacun un nom et un président dont on donnera le nom et le prénom.
  - Plusieurs équipes peuvent venir du même club. Elles ont chacune un nom et un capitaine d'équipe qui est joueur de cette équipe.
  - Chaque joueur appartient à une et une seule équipe, il est caractérisé par son nom, prénom et numéro de licence.
  - Chaque match est défini par un numéro unique. Pour chaque match, les deux équipes et le score sont enregistrés. On partira du principe qu'il n'y a pas de match nul.
1. Etablir un modèle entités-associations.
  2. Transformer ce modèle entités-associations en un modèle relationnel.
  3. Ecrire en SQL les requêtes suivantes :
    - a. Inscrire la première équipe. Son nom est *Matu* et son capitaine se nomme *Lagagne*. Le club dont fait partie l'équipe se nomme *Rite*.
    - b. Ajouter à la table adéquate le champ *Adresse* permettant d'ajouter l'adresse de chacun des clubs.
    - c. Rechercher combien de joueurs se prénomment *Jean*.
    - d. Rechercher dans quelle équipe joue *Serge Lagagne*.

### Exercice 3 : Arbre binaire de recherche (14 points)

1. Soit un ABR donné sous la forme d'une liste : [8 ; 3 ; 10 ; 1 ; 6 ; - ; 14 ; - ; - ; 4 ; 7 ; - ; - ; 13 ; -].
  - a. Que donne cette liste lorsqu'on y ajoute le nombre 12 ?
  - b. Que donne la liste initiale lorsqu'on lui enlève le 3 ?
2. Combien y a-t-il de formes d'arbres binaires différents à 4 nœuds ? Les représenter graphiquement sans donner de valeur aux nœuds. Parmi ceux-ci, combien y en a-t-il qui sont entiers ? parfaits ?

Indication : voici une forme possible à 3 nœuds



3. Comment trouver la valeur la plus grande dans un arbre binaire de recherche ?
4. Les séquences suivantes peuvent-elles correspondre à la recherche de 108 dans un arbre binaire de recherche ?
  - a. 10, 15, 14, 150, 120, 108
  - b. 10, 15, 150, 14, 120, 108
  - c. 10, 14, 150, 15, 120, 108
  - d. 120, 150, 14, 15, 10, 108
  - e. 150, 10, 14, 15, 120, 108

### Exercice 4 : Codage (12 points)

1. Soit le code : 1-46619-909-3.
  - a. Montrer que ce code n'est pas un code ISBN-10 correct.
  - b. On imagine qu'il n'y a qu'une erreur. Est-il possible que l'erreur se situe dans le deuxième chiffre ? Si oui, corriger ce chiffre afin d'obtenir un code ISBN-10 correct.
2. Convertir  $(213, \bar{4})_{10}$  en base 2.

### Exercice 5 : Python (20 points)

1. En utilisant le langage *Python 3*, créer une classe *TelephonePrePaye* qui permette de gérer financièrement l'utilisation d'un téléphone portable capable uniquement de téléphoner. Il sera défini par un nom de client et un solde (nombre de francs encore à disposition). Les méthodes de cette classe devront permettre d'afficher le solde, de déposer de l'argent et de décompter CHF 1.- par communication (tarif unique). S'il n'y a plus d'argent sur le compte le message « Solde insuffisant » devra s'afficher.  
Définir une classe *Enfant* identique à la première, mise à part qu'il n'est pas possible de déposer plus de CHF 20.- à la fois. Cette classe héritera de tous les éléments possibles de la classe *TelephonePrePaye*.
2. Soit un nombre entier  $a$ . Ordonner à l'aide de *Python 3* les chiffres de  $a$  du plus petit au plus grand. Par exemple, au nombre 5682342, le programme fera correspondre le nombre 2234568. L'utilisation de la méthode *sort()* n'est pas autorisée dans cet exercice.