



Session 2011

Examen OC INFORMATIQUE

Informations et consignes :

- Temps à disposition : 3 heures
- Formulaire annexé autorisé
- Calculatrice non programmable autorisée
- Note maximale (6) pour 5 problèmes justes
- Tous les problèmes ont le même poids.
- Vous devez écrire proprement au stylo ou à l'encre. La présentation est prise en compte pour la note.
- Utilisez une feuille par problème.
- Écrivez votre nom sur chaque feuille.
- Faites une marge de 2 cm à gauche.
- Rendez tous vos documents, y compris la donnée et le formulaire.

Problème 1 : Divers

- a) Une alarme acoustique est installée dans une voiture. Celle-ci s'enclenche si :
- la clé de contact est sur *ON*, il y a un poids sur le siège conducteur et la ceinture n'est pas bouclée **ou** si
 - la clé de contact est sur *ON* et il n'y a aucun poids sur le siège.
1. Écrire la fonction logique simplifiée mettant l'alarme en marche.
2. Construire le circuit logique correspondant.
- b) Transformer le nombre $(-29)_{10}$ de base 10 en base 2 (en complément à 2).
Transformer le nombre $(29,5625)_{10}$ en base 2.
- c) En une phrase, qu'est-ce que le *BIOS* ?
- d) Dans le cadre d'une retouche d'image, comment diminuer la luminosité sans saturation ?
Faire un schéma.

Problème 2 : Arbre binaire de recherche

Dans le cadre des arbres binaires de recherche (ABR), définissons une classe *ArbreBinaire()*. Les instances de cette classe sont les nœuds de l'arbre. Parmi les attributs et méthodes de cette classe nous trouvons :

abr.fils_gauche : pointe vers le fils gauche, nul par défaut
abr.fils_droit : pointe vers le fils droit, nul par défaut
abr.valeur : donne la valeur du nœud, nul par défaut
abr.racine : pointe vers la racine de l'arbre
abr.insere(valeur) : range le nombre *valeur* dans l'arbre

Soit *A* un arbre vide. Les nombres 6, 2, 10, 1, 8 sont insérés dans cet ordre à l'intérieur de l'ABR.

- a) Dessiner l'arbre *A* après l'insertion de ces nombres.
- b) Qu'est-ce qu'un arbre AVL ? L'arbre obtenu ci-dessus est-il un arbre AVL ?
- c) Donner la liste des valeurs visitées lors d'un parcours *préfixé*, *infixé*, ou *postfixé*.
- d) Quel est l'avantage d'utiliser un ABR par rapport à une liste ?
- e) Écrire l'algorithme *insere(valeur)* en pseudo-code. On considérera ici qu'il ne peut y avoir deux fois la même valeur. On imaginera que le premier nœud est déjà défini.
- f) Quelle est la complexité dans le pire des cas pour la recherche ?

Problème 3 : Python – simulation du loto

- a) Écrire une fonction *jeu()* en Python 2.6 qui permet à un joueur d'entrer 6 nombres entiers, chacun variant entre 1 et 49. On suppose que l'utilisateur entre des nombres entiers. On vérifiera que les valeurs sont correctes et différentes. Les nombres inscrits par le joueur seront placés dans une liste renvoyée par la fonction.
- b) Écrire une fonction *tirage()* dans le même langage qui renvoie une liste de 6 nombres entiers aléatoires différents entre 1 et 49. Cette liste correspond au tirage du loto.
- c) Un jeu est gagnant si au moins 3 nombres introduits par le joueur sont inclus aussi dans le tirage.
- d) Écrire une fonction *gagnant(jeu, tirage)* également en Python 2.6, qui renvoie « gagné » si le jeu inscrit par le joueur est gagnant et « perdu » dans le cas contraire.

Problème 4 : Base de données

Une nouvelle entreprise d'assurance « Mabil » se lance dans les assurances de voitures. Pour gérer ses clients elle décide de créer une base de données.

Chaque client peut assurer plusieurs voitures. Les assurances (et les sinistres) sont liées aux véhicules.

- a) Établir un modèle entités-associations.
- b) Transformer votre modèle entités-associations en un modèle relationnel.
- c) Écrire en SQL les requêtes suivantes :
 1. Ajouter à la table voiture le champ contrat permettant d'ajouter la date du début du contrat.
 2. Rechercher combien l'assurance « Mabil » possède de voitures assurées.
 3. Rechercher quelles sont les voitures assurées par « Jean Cerf ». On suppose qu'une seule personne assurée porte ce nom.
 4. Rechercher le nombre total d'accidents qu'ont eus les voitures de « Jean Cerf ».

Problème 5 : Codage

- a) Code de Hamming.
 1. Un code de Hamming a 5 bits de contrôle. Quelle est la longueur de ce code ? Où se trouvent ces bits de contrôle ?
 2. Coder le message 111 1110 0000.
 3. Décoder et corriger si nécessaire le message 110 1001.
- b) Code CRC.

Connaissant le polynôme générateur $x^3 + x + 1$, calculer le code du message 1100 0110 1001.