



Filière : Techniques de Développement Informatique

Niveau: TS

Durée : 5 heures

Epreuve : Synthèse

Variante : V1

Barème : / 120Pts

❖ **Partie I : Théorie (40 pts)**

➤ **Dossier 1: Notions de mathématiques appliquées à l'informatique (12 pts)**

▪ **Exercice 1: (6 pts)**

NB : la calculatrice est strictement interdite.

Compléter le tableau suivant:

Décimal	Binaire	Octal	Hexadécimal
1796			
	10101100		
		1767	
			E10

▪ **Exercice 2: (6 pts)**

Soit la fonction logique suivante:

$$Q(k, l, m, n) = \overline{\overline{m}ln} + \overline{mk\overline{l}} + lm\overline{n}$$

- 1- Simplifier analytiquement la fonction logique Q . (2 pts)
- 2- Construire la table de vérité. (2 pts)
- 3- Simplifier avec la méthode de Karnaugh la fonction logique Q . (2 pts)

➤ **Dossier 2: Techniques de programmation structurée (16 pts)**

▪ **Exercice 1: Vérification d'un tableau trié (8 pts)**

Ecrire une fonction "*est_trié()*" permettant de vérifier si un tableau de réelles T de taille N (passés en argument de cette fonction) est trié ou non. La fonction doit renvoyer l'un des trois messages: « *trié croissant* », « *trié décroissant* » ou « *non trié* ». Faites appel à cette fonction dans un algorithme donnant le droit à l'utilisateur de remplir un tableau avec N valeurs.

Filière	Epreuve	Session	1/7
TDI	Synthèse V1	Juillet 2014	

Exemple:

Donner le nombre des éléments du tableau:

4

Donner les éléments du tableau:

-15.33

-8.54

-3.68

19.75

Votre tableau est trié croissant.

▪ **Exercice 2: Le cerf-volant (8 pts)**

Ecrire une procédure "*cerf_volant()*" permettant de dessiner un cerf-volant dont la taille N est passée en paramètre de la procédure. Faites appel à cette procédure dans un algorithme donnant le droit à l'utilisateur de saisir la taille du dessin qui doit être paire.

Exemple:

Donner la taille paire:

3

La taille doit être paire!

Donner la taille paire:

8



Dossier 3: Conception et modélisation d'un système d'information (12 pts)

Conception d'un système d'information pour une société informatique

Une grande société de fabrication de produits informatiques souhaite réaliser un site web, permettant de présenter ses produits à la clientèle. La société a plusieurs filiales installées dans plus d'une cinquantaine de pays à travers le monde, réparties sur 8 zones géographiques: Amérique du nord, Amérique latine, Europe, Afrique, Moyen-Orient, Asie, Extrême-Orient et Australie. Dans sa politique commerciale, la société propose pour chaque pays un ensemble de produits s'adaptant au pouvoir d'achat des citoyens du pays.

L'interface du site web permet de rediriger automatiquement un internaute sur la page de son pays pour avoir accès à la liste des produits disponibles dans son pays. Les produits quant à eux sont répertoriés sous plusieurs catégories (par exemple: Ordinateurs, Imprimantes,...). On choisissant la catégorie "Ordinateurs" par exemple, l'intéressé sera appelé à choisir entre "Postes de travail", "Ordinateurs portables",.... Par contre, il y a des catégories comme "Scanners" où l'utilisateur se voit directement devant la liste de produits sans avoir à passer par un deuxième choix.

La société donne aux clients de certains pays le droit d'effectuer des achats directement à partir du site. Pour effectuer un achat, le client doit s'enregistrer dans le site en fournissant son nom et prénom, son adresse, sa ville (à partir d'une liste déroulante), son code postal (généralisé automatiquement après sélection

Filière	Epreuve	Session	2/7
TDI	Synthèse VI	Juillet 2014	

de la ville), son numéro de téléphone et son e-mail. L'enregistrement ne se valide sauf si le client achète au moins un produit, en spécifiant la quantité achetée du produit.

- 1- Etablir le dictionnaire de données. (3 pts)
- 2- Etablir le modèle conceptuel de données. (6 pts)
- 3- Etablir le modèle logique de données. (3 pts)

❖ Partie II: Pratique (80 pts)

➤ Dossier 1: Langage de programmation structurée (20 pts)

▪ Exercice 1: (10 pts)

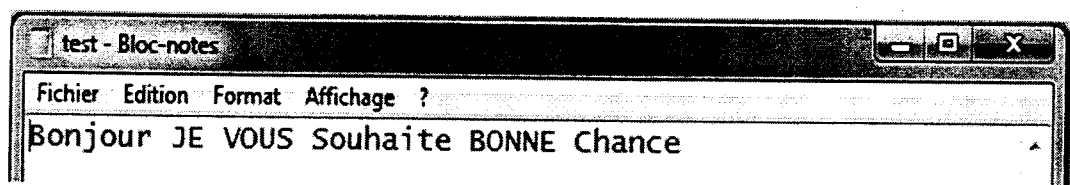
Soit le programme C suivant :

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    FILE *fpi=fopen("c:\\ofppt\\test.txt","r");
    FILE *fpo=fopen("c:\\ofppt\\res.txt","w");
    int c;
    for(;(c=fgetc(fpi)) != EOF;){
        if(c == ' ') //test sur le caractère espace
            fputc('\n', fpo);
        else
            fputc(((toupper(c) + 5 - 'A') % 26) + 'A' , fpo);
    }
    getchar();
}
```

NB :

- i. ***toupper** c'est une fonction de la bibliothèque **ctype.h** qui convertit un caractère en majuscule.*
- ii. *Les codes ASCII des caractères de 'A' à 'Z' sont ordonnés. Si le code de 'A' est une valeur **a** alors celui de 'B' est **a+1**,... celui de 'Z' est **a+25**.*

Le fichier « test.txt » contient le texte suivant :



- a. Que fournit l'expression : « ((toupper(c) + 5 - 'A') % 26) + 'A' » pour les valeurs suivantes de **c** : 'B', 'L', 'V' et 'Z'. C'est quoi donc le rôle de cette expression. (4 pts)
- b. Quel est le contenu de fichier « res.txt » après exécution du programme. (6 pts)

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	3/7
TDI	Synthèse V1	Juillet 2014	

- **Exercice 2 : (10 pts) :** Quel est le résultat d'affichage du programme suivant

```
#include <stdio.h>
int gct(int *k)
{
    int x=6;
    printf("Debut gct : x=%d, *k=%d\n", x,*k); //2 pts
    x=*k+1;
    printf("fin gct: x=%d, *k=%d\n",x, *k); // 2 pts
    return(x * *k);
}

void fct(int n, int *m)
{
    printf("Debut fct n=%d, *m=%d\n", n, *m); //1 pt
    n=*m+3;
    *m=gct(&n);
    printf("Fin fct n=%d, *m=%d\n", n, *m); //2 pts
}

int main()
{
    int a=5, b=3;
    printf("Debut main a=%d, b=%d\n", a, b); //1 pt
    fct(a,&b);
    printf("Fin main a=%d, b=%d\n", a, b); //2 pts
    getchar();
}
```

➤ **Dossier 2: Programmation orientée objet (35 pts)**

Développement d'une application orientée objet pour la société informatique

On souhaite développer une application orientée objet à l'instar du système conçu pour la société de fabrication des produits informatiques. L'application contiendra plusieurs classes dont tous les attributs doivent être **privés**, ce qui implique le besoin de coder des **accesseurs** dans toutes les classes. Les accesseurs des attributs affectées automatiquement doivent être en **lecture seule**. Chaque classe doit avoir un **constructeur d'initialisation**, et une méthode "ToString".

On suppose que les classes suivantes sont déjà créées :

- **Classe "Zone"** : Ayant comme attributs un identifiant pour la zone géographique (entier positif qui s'incrémente automatiquement), et le nom de la zone géographique

```
public class Zone
{
    private uint IdZone;
    private string NomZone;
    private static uint auto;
}
```

- **Classe "Pays"** : Ayant comme attributs un identifiant pour le pays (entier positif qui s'incrémente automatiquement), le nom du pays, et sa zone géographique.

```
public class Pays
{
    private uint IdPays;
    private string NomPays;
    private Zone zone;
    private static uint auto;}
```

Filière	Epreuve	Session	4/7
TDI	Synthèse V1	Juillet 2014	

- **Classe "Ville"** : Ayant comme attributs un identifiant pour la ville (entier positif qui s'incrmente automatiquement), le nom de la ville, son code postal (entier positif) et le pays où se situe la ville.

```
public class Ville
{
    private uint IdVille;
    private string NomVille;
    private uint CodePostal;
    private Pays pays;
    private static uint auto;}
```

- **Classe "Catégorie"** : Ayant comme attributs un identifiant pour la catégorie (entier positif qui s'incrmente automatiquement), le nom de la catégorie, et la catégorie parente, et un 2^{ème} constructeur avec un seul paramètre (nom de la catégorie).

```
public class Catégorie
{
    private uint IdCatégorie;
    private string NomCatégorie;
    private Catégorie catégorie;
    private static uint auto;}
```

- **Classe "Produit"** : Ayant comme attributs un identifiant du produit (entier positif qui s'incrmente automatiquement), le nom du produit, son prix et sa catégorie.

```
public class Produit
{
    private uint IdProduit;
    private string NomProduit;
    private float PrixProduit;
    private Catégorie catégorie;
    private static uint auto;}
```

Vous devez répondre aux questions suivantes :

- 1- Dans la classe "Pays", ajouter un nouvel attribut "**Produits_disponibles**" qui sera une liste générique des produits disponibles dans le pays. (2,5 pts)
 - 2- Dans la classe "Pays", ajouter une méthode "**AjouterProduit(produit)**" permettant d'ajouter un nouveau produit à la liste des produits disponibles dans le pays. Une exception est levée si le produit est déjà existant (produit ayant le même identifiant). (3,5 pts)
 - 3- Dans la classe "Pays", ajouter une méthode "**SupprimerProduit(identifiant_produit)**" permettant de supprimer un produit en fournissant son **identifiant**. Une exception est levée si l'identifiant du produit est inexistant. (3,5 pts)
 - 4- Dans la classe "Pays", ajouter une méthode "**string ListerProduits()**" permettant de lister les produits disponibles dans le pays. (2,5 pts)
- **Classe "Achat"**
 - 5- Créer une classe "Achat" ayant comme attributs le produit acheté, la quantité achetée, et la date d'achat (affectée automatiquement par la date système). (3,5 pts)
 - 6- Ajouter une méthode "**MontantAchat()**" permettant de calculer le montant d'un achat. (2,5 pts)

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	5/7
TDI	Synthèse V1	Juillet 2014	

▪ Classe "Client"

- 7- Créer une classe "Client" ayant comme attributs un identifiant du client (entier positif qui s'incrémente automatiquement), son nom et son prénom, son adresse postale, sa ville (de type "ville") son numéro de téléphone, son adresse e-mail et la liste des achats qu'il a effectués. (4 pts)
- 8- Ajouter une méthode "AcheterProduit(achat)" permettant d'ajouter un achat à la liste des achats du client. (3 pts)
- 9- Ajouter une méthode "string ListerAchats(date_début, date_fin)" permettant de lister les achats effectués par le client entre une date de début et une date de fin. (4 pts)

▪ Classe "Client Fidèle"

Un client fidèle "est un client" qui s'adhère au système de fidélisation de la société en payant un montant annuel pour l'abonnement, ce qui lui permet de bénéficier d'un numéro de fidélité et avoir une réduction de 25% sur ses achats.

- 10- Créer la classe "Client_Fidèle" ayant comme attributs un numéro de fidélité (entier positif qui s'incrémente automatiquement). (3 pts)
- 11- Redéfinir la méthode "AcheterProduit(achat)" permettant d'ajouter un achat à la liste des achats du client après avoir appliqué une réduction de 25% sur le prix du produit acheté. (3 pts)

➤ Dossier 3: Programmation événementielle (25 pts)

NB : Dans ce dossier, on vous demande de donner uniquement le code à mettre à l'intérieur des méthodes événementielles. L'entête de ces méthodes événementielles n'est pas demandé !

Soit le formulaire suivant nommée "Liste des achats":

	Nom Produit	Quantité_Achetée	Date Achat	Montant Achat
	NetBook	2	20/01/2014 15:46:23	5980 DH
	NoteBook	2	20/01/2014 15:46:23	9980 DH
➤*				

- 1- Dans le code du formulaire on a déclaré un objet de type "Client":

```
Client c1;
```

Dans la méthode de chargement du formulaire on a instancié l'objet "Client":

```
c1 = new Client("ALAMI", "Ahmed", "25, Rue Riad, Qu. Oumnia", rabat, "0689741211", "alami@mail.ma");
```

Filière	Epreuve	Session	6/7
TDI	Synthèse V1	Juillet 2014	

Ajouter dans la méthode de chargement du formulaire, le code permettant d'afficher l'en-tête du formulaire comme ce qui est indiqué dans l'image ci-dessus. (5 pts)

- 2- Ecrire le code du bouton "**Afficher**" permettant d'afficher la liste des achats du client "c1", effectuées entre une date de début et une date de fin, sélectionnées par l'utilisateur (voir image ci-dessus). (8 pts)
- 3- Ecrire le code du bouton "**Sauvegarder**" qui permet d'enregistrer la liste des achats du client "c1", effectuées entre une date de début et une date de fin, sélectionnées par l'utilisateur (voir image ci-dessus) dans un fichier *objet* (sérialisation) qui porte le nom « AchatsAlami.dat ». (8 pts)
- 4- Ecrire le code du bouton "**Effacer**" permettant d'effacer le contenu du composant graphique représentant la liste, et d'initialiser les sélectionneurs des dates à la date système. (4 pts)

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	7/7
TDI	Synthèse VI	Juillet 2014	