

**Examen de Fin de Module, Formation Initiale**  
**Année 2014 -2015**  
**SESSION JANVIER 2015**

**Filière :** TDI

**Niveau :** TS

**Durée :** 2 H 00

**Barème :** 40 points

**Intitulé du module :** Langage de programmation structurée

**Partie I : Théorique (11 pts)**

1. C'est quoi le rôle d'un compilateur ? **(2 pts)**
2. Quelle est la différence entre le passage de paramètres d'une fonction par adresse et le passage de paramètres par valeur ? donner un exemple. **(3 pts)**
3. Citer quatre fonctions de la bibliothèque **string.h**. **(2 pts)**
4. Que sera le résultat d'exécution des programmes suivants : **(4 pts)**

Programme 1	Programme 2	Programme 3
<pre>int i = 10; for(i = 0; i &lt; 10; i++)     printf("%d\n", i);</pre>	<pre>int i = 0; while(i &lt; 10);     i++;     printf("%d\n", i);</pre>	<pre>int i = 0, j = 10; do{     printf("%d\n", i);     i++;     j = i + j; }while((i &lt; 10) &amp;&amp; (j &lt; 20));</pre>

Programme 4
<pre>int i = 1; switch(i){     case 0 : puts("0");     case 1 : puts("1");     case 2 : puts("2");     default : puts("Good By"); }</pre>

**Partie II : Pratique (29 pts)**

**Exercice 1 : (5 pts)**

On considère la suite numérique définie par :

$$U_1 = 1$$

$$U_2 = 2 \text{ et}$$

$$U_n = 2 (U_{n-1} + U_{n-2}) \text{ (pour } n > 2)$$

1. Ecrire une fonction qui retourne, le  $n^{\text{ième}}$  terme de cette suite ; **(3 pts)**
2. Ecrire un programme qui calcule et affiche les termes, de cette suite, inférieurs ou égaux à un entier entré par l'utilisateur ; ainsi que leur somme. **(2 pts)**

## Exercice 2 : (24 pts)

On souhaite créer un programme en langage C permettant de gérer les articles dans un stock informatique. Un Article est caractérisé par les champs **code** (entier), **nom** et **prixUnitaire**.

1. Définir la structure **Article**. (2 pts)
2. Créer une procédure **void lireArticle (Article articles [], int n)** permettant de remplir un tableau de n articles. (3 pts)
3. Créer une procédure **void afficherArticle (Article articles [], int n)** permettant d'afficher la liste des articles stockés dans le tableau **articles**. (2 pts)
4. Créer une procédure **void supprimerArticle (Article articles [], int \*n, int code)** permettant de supprimer un article stockés dans le tableau **articles** en fournissant son code. (3 pts)
5. Créer une procédure **void ajouterArticle (Article articles [], int \*n)** permettant d'ajouter un article dans le tableau **articles**. (2 pts)
6. Créer une procédure **void triByNom (Article articles [], int n)** permettant d'afficher les articles classés par ordre alphabétique des noms. (2 pts)
7. Créer une fonction **Article \* triByPrix (Article articles [], int n)** permettant de renvoyer un tableau d'articles classés du plus cher au moins cher. (2 pts)
8. Créer une procédure **void saveArticles (Article articles [], int n)** permettant d'enregistrer les articles dans un fichier texte, les valeurs des champs de chaque article seront séparées par « ; ». (3 pts)
9. Ecrire un algorithme qui affiche à l'utilisateur le menu suivant : (5 pts)
  - 1: Lire les articles.
  - 2: Afficher les articles classés par ordre alphabétique des noms.
  - 3: Afficher les articles classés du plus cher au moins cher.
  - 4: Ajouter un nouveau article.
  - 5: Supprimer un article.
  - 6: Enregistrer les articles.
  - 7: Quitter le programme.Tapez votre choix :

et réalise l'opération demandée par l'utilisateur en utilisant les fonctions définies précédemment.