

Filière : TDI

Niveau : 1^{ère} année

Durée : 2h

Intitulé du module : Langage de programmation structurées

Barème: ... / 40

I-Partie Théorique: (4 pts)

1. Citer 6 fonctions de manipulation des chaînes de caractères de la bibliothèque **string.h** (3 pts)
2. Qu'est ce qu'il représente le nom d'un tableau lors d'un appel de fonction, du genre
int n = Somme(nom_tableau , n) (1 pt)

II-Partie Pratique : (38 Points)

Exercice 1 : (8 pts)

On considère la suite numérique définie par : $S_1=1$
 $S_2=2$ et
 $S_n = (S_{n-1} \times S_{n-2})$ (pour $n > 2$)

- a. Ecrire une fonction récursive qui retourne, le n^{ième} terme de cette suite ; (3pt)
- b. Ecrire une fonction qui calcule et affiche les termes, de cette suite, inférieurs ou égaux à un entier passé en paramètre ainsi que leur produit, si le n passé n'est pas définie pour la suite on affiche un message d'erreur (3pt)
- c. Ecrire un programme de test pour la question a et b (2pt)

Problème : (28 pts)

Soit la structure nommée Salle caractérisée par son identificateur et son libellé, et le champ capacité qui représente le nombre des étudiants maximal.

1. Créer la structure Salle (1pt)
2. Définir la fonction **Salle CreerSalle(int idSalle,char *libelleSalle,int Capacite)** qui permet de créer une salle et de la retourner (2pt)
3. Définir la fonction **void SaisirInfosSalle(Salle Liste[],int n)** qui permet d'insérer n Salle dans un tableau de stockage des salles passée en paramètre utilisant la question précédente. N°2 (3pt)

4. Définir la fonction **void AfficherSalle(Salle S)** qui permet d'afficher les informations de la salle passé en paramètre sous la forme :
Id Salle : ; Libellé Salle : ; Capacité Salle : (2pt)
5. Définir la procédure **void AfficherInfosSalles(Salle Liste[], int n)** permettant d'afficher les informations des salles utilisant la fonction de la question N° 4. (3pt)
6. Définir la fonction **bool RechercherSalle(int id,Salle Liste[],int n)**
Qui permet de rechercher une salle par son identificateur passé en paramètre, le parcours doit s'arrêter une fois la salle est trouvée. (3pt)
7. Définir la fonction **void SupprimerSalle((int id,Salle Liste[],int *n)**
Permettant de supprimer la salle du tableau passé en paramètre utilisant la question N°6, si la salle à supprimer n'existe pas on affiche un message d'erreur. (3pt)
8. Définir la fonction **void TrierSallesParLibelle(Salle Liste[], int n)**
Qui permet de Trier en ordre croissant des libellés salle, et d'afficher leurs informations utilisant la fonction de la question N°5. (3pt)
9. Définir la fonction **void Sauvegarder (Salle Liste[], int n)**
Qui permet de sauvegarder les informations des différentes salles dans un fichier nommée Salles.txt (3pt)
10. Créer un menu de Test qui utilise les fonctions des questions N° : 3 , 5 , 7 , 8 , 9 (4pts)

Bon Courage