

Devoir surveillé <input type="checkbox"/>	Examen <input checked="" type="checkbox"/>	Session : principale <input type="checkbox"/> de contrôle <input checked="" type="checkbox"/>
Matière : Algorithmique SD 1		Semestre: 1
Enseignant : Wided Miled et Aymen Sellaouti		Date: 17/06/2017 à 10h00
Filière(s) : MPI		Durée: 1h30
Nombre de pages : 2		Documents : autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/>

### Exercice 1 (4 points) :

Ecrire en algorithmique une procédure qui demande à l'utilisateur un entier  $n$  et affiche les  $n$  premières lignes du *triangle de Pascal*. On rappelle que le triangle de Pascal est une façon de présenter et de calculer les coefficients binomiaux  $C_j^i$ . Dans chaque ligne  $i$  et colonne  $j$  du triangle, est placé le coefficient binomial  $C_j^i$  qui satisfait les relations

$$\begin{cases} \forall i, j | 0 < j < i, C_j^i = C_{j-1}^{i-1} + C_j^{i-1} \\ C_0^i = C_i^i = 1 \end{cases}$$

Exemple : le triangle de pascal de rang 5 :

```

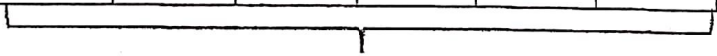
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
    
```

### Exercice 2 (4 points)

On se propose d'écrire un algorithme permettant de vérifier si un tableau est symétrique d'ordre 3 ou non. Un tableau est dit symétrique de degré 3 s'il existe au moins 3 cases consécutives dont la somme est égale à la somme des 3 cases suivantes.

Exemple: tableau de symétrie 3

1	3	6	3	2	5	3	3	1	1
---	---	---	---	---	---	---	---	---	---



Symétrie de degré 3

### Exercice 3 (12 points) :

Un gérant d'un ensemble de terrain de football souhaite commander une application pour la gestion des réservations des terrains. Chaque réservation concerne un mois complet. Le prix de la location n'est pas fixe et il est négocié avec le client à chaque réservation.

Le fichier « reservation.txt » a la structure suivante :

Code\_reservation : 10 caractères  
Num\_terrain : 4 caractères numériques

Mois : 2 caractères numériques  
 Année : 2 caractères numériques  
 Prix : 4 caractères numériques  
 CINClient : 8 caractères numériques

Soit la structure modélisant une réservation.

```
typedef struct reservation{
  char codeRes[11];
  int numTerrain;
  int mois;
  int annee;
  int prix;
  long CINClient;
}Resa;
```

Chaque ligne du fichier représente une réservation d'un terrain pour un mois.

- 1- Ecrire la fonction `int loadFile (Resa t[ ])` qui permet de charger le contenu du fichier « reservation.txt » dans un tableau de structure, `n` étant le nombre de cases remplies.
- 2- Ecrire la fonction `int totaleRecetteMois (Resa t[ ], int n, int mois, int annee)` qui retourne la recette totale des réservations pour un mois donné.
- 3- Ecrire la fonction `int meilleurMois (Resa t[ ], int n, int annee)` qui permet de retourner le mois ayant la meilleure recette d'une année donnée.
- 4- Afin d'identifier les clients les plus fidèles, on se propose de créer la structure Client suivante :

```
typedef struct Client{
  long CINClient;
  int totalPaiement;
}Client;
```

Le champ `totalPaiement` représente le total des paiements effectués par le client pour la totalité de ses réservations.

Ecrire la fonction `int loadClient (Resa t[ ], int n, Client tClient[ ])` qui permet de générer, à partir du tableau `Resa`, le tableau `tClient` contenant l'ensemble des clients.

Un client ne peut apparaître qu'une seule fois dans le tableau `tClient`.

Vous n'avez le droit qu'à un seul parcours.

Bon Travail