

**Exercice 1 :** En mathématiques,  $C_n^p$  désigne le nombre de sous-ensembles à  $p$  éléments dans un ensemble à  $n$  éléments. Les nombres  $C_n^p$  sont appelés coefficients binomiaux car ils apparaissent comme coefficients dans une formule célèbre, dite du binôme de Newton.

Pour tous entiers  $n$  et  $p$

- si  $p > n$ ,  $C_n^p = 0$ ,
- $C_n^0 = 1$ ,
- $C_n^n = 1$ ,
- si  $1 \leq p \leq n - 1$ , on dispose d'une formule pour calculer  $C_n^p$  :  

$$C_n^p = C_{n-1}^{p-1} + C_{n-1}^p$$

**Question 1.1 :** Ecrire une fonction php qui prend en paramètre un entier  $k$  et qui renvoie un tableau  $t$  tel que pour tous  $n$  et  $p$  inférieurs ou égaux à  $k$ ,  $t[n][p]$  contient le coefficient  $C_n^p$ .

**Question 1.2 :** Ecrire une fonction php qui prend en paramètre un tableau, résultat de la fonction précédente avec un paramètre  $k$  et renvoie sa représentation sous la forme d'une table HTML à  $k + 1$  lignes et  $k + 1$  colonnes (le résultat de la fonction est donc une chaîne). Par exemple, le code

```
$t = creerTabCoef(10);
echo afficherTabCoef($t) ;
```

Entraîne la génération de la table HTML suivante (figure ci-dessous, à gauche) :

|   |    |    |     |     |     |     |     |    |    |   |
|---|----|----|-----|-----|-----|-----|-----|----|----|---|
| 1 |    |    |     |     |     |     |     |    |    |   |
| 1 | 1  |    |     |     |     |     |     |    |    |   |
| 1 | 2  | 1  |     |     |     |     |     |    |    |   |
| 1 | 3  | 3  | 1   |     |     |     |     |    |    |   |
| 1 | 4  | 6  | 4   | 1   |     |     |     |    |    |   |
| 1 | 5  | 10 | 10  | 5   | 1   |     |     |    |    |   |
| 1 | 6  | 15 | 20  | 15  | 6   | 1   |     |    |    |   |
| 1 | 7  | 21 | 35  | 35  | 21  | 7   | 1   |    |    |   |
| 1 | 8  | 28 | 56  | 70  | 56  | 28  | 8   | 1  |    |   |
| 1 | 9  | 36 | 84  | 126 | 126 | 84  | 36  | 9  | 1  |   |
| 1 | 10 | 45 | 120 | 210 | 252 | 210 | 120 | 45 | 10 | 1 |

|   |    |    |     |     |     |     |     |    |    |   |
|---|----|----|-----|-----|-----|-----|-----|----|----|---|
| 1 |    |    |     |     |     |     |     |    |    |   |
| 1 | 1  |    |     |     |     |     |     |    |    |   |
| 1 | 2  | 1  |     |     |     |     |     |    |    |   |
| 1 | 3  | 3  | 1   |     |     |     |     |    |    |   |
| 1 | 4  | 6  | 4   | 1   |     |     |     |    |    |   |
| 1 | 5  | 10 | 10  | 5   | 1   |     |     |    |    |   |
| 1 | 6  | 15 | 20  | 15  | 6   | 1   |     |    |    |   |
| 1 | 7  | 21 | 35  | 35  | 21  | 7   | 1   |    |    |   |
| 1 | 8  | 28 | 56  | 70  | 56  | 28  | 8   | 1  |    |   |
| 1 | 9  | 36 | 84  | 126 | 126 | 84  | 36  | 9  | 1  |   |
| 1 | 10 | 45 | 120 | 210 | 252 | 210 | 120 | 45 | 10 | 1 |

**Question 1.3 :** Que faire pour que les cases vides n'apparaissent pas, tout en conservant une table à  $k + 1$  lignes de  $k + 1$  colonnes (figure ci-dessus, à droite) ?

**Exercice 2 :** Ecrivez une fonction `sortieTable` qui en prend en paramètre une table d'association et qui produit une table HTML dont chaque ligne représente un couple (`clef`, `valeur`). Cette fonction sera récursive car si une valeur de la table d'association est elle-même une table alors il faut produire pour cette valeur une sous-table HTML.