

INF233-TP1 : Énumérer et Compter

NB : Les questions marquées d'une ou plusieurs étoiles sont plus compliquées. Elles ne sont à traiter que lorsque le reste est fait.

Exercice 1 : Calcul de la factorielle et des coefficients binomiaux

1. Ecrire une fonction `int factorielle(int n)` qui renvoie la valeur $n!$. Donner une version itérative et récursive de cette fonction. Tester votre fonction.
2. Ecrire une fonction `int coefficientbinomiaux(int n, int k)` qui calcule la valeur $\binom{n}{k}$ (vous pouvez utiliser la fonction précédente, qui n'est pas la manière optimale de procéder).
3. Afficher le triangle de Pascal sur les 6 premières lignes (là encore, vous pouvez utiliser la fonction précédente, même si ce n'est pas optimal). Le résultat obtenu pourra ressembler à :

```
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
```

Exercice 2 : Fibonacci

La suite de Fibonacci $(F_n)_{n \in \mathbb{N}}$ est définie par récurrence de la manière suivante :

- $F_0 = 1$
- $F_1 = 1$
- Pour $n > 0$, $F_{n+1} = F_n + F_{n-1}$

Écrire une fonction qui calcule le n -ième terme de la suite de Fibonacci. Donner une version récursive et itérative. Comparer les deux fonctions : laquelle est la plus rapide (en terme de nombre d'opérations) ?

Exercice 3 : Somme d'entiers

Dans chacun des cas suivants, écrire une fonction qui prend un entier n en argument et qui calcule la somme de :

- tous les entiers de 1 à n ,
- tous les entiers impairs de 1 à n ,
- tous les carrés des entiers de 1 à n .

Dans chacun des cas, tester votre fonction et essayer de deviner une formule qui donne la valeur de la somme en fonction de n . (Certaines formules seront vues et démontrées en TD).

Exercice 4 : Sous-ensemble à k éléments de $\{1, \dots, n\}$

1. Ecrire à la main tous les sous-ensembles (non ordonnés) à deux éléments de $\{1, \dots, 5\}$.
2. Ecrire une fonction qui prend un entier n en argument et qui affiche la liste de tous les sous-ensembles à deux éléments de $\{1, \dots, n\}$. Modifier votre fonction pour qu'elle renvoie le nombre de sous-ensembles affichés. Deviner une formule pour ce nombre.
3. Faire de même avec les sous-ensembles à trois éléments de $\{1, \dots, n\}$.
4. Faire de même (question 3) pour les sous-ensembles ordonnés à trois éléments de $\{1, \dots, n\}$.
5. (***) Ecrire un programme qui énumère les sous-ensembles à k éléments de $\{1, \dots, n\}$.

Exercice 5 : Mots de longueur n sur un alphabet à k lettres

Rappel : Pour écrire la i -ème lettre de l'alphabet on pourra utiliser la commande `putchar('a'+i-1);`.

1. Ecrire à la main tous les mots de quatre lettres sur l'alphabet $\{a, b\}$.

2. Ecrire une fonction qui prend un entier k en argument et qui affiche la liste de tous les mots de quatre lettres utilisant les k premières lettres de l'alphabet. Modifier votre fonction pour qu'elle renvoie le nombre de mots affichés. Deviner une formule pour ce nombre.
3. (*) Modifier le programme précédent pour qu'il n'affiche que les mots contenant au plus deux fois la lettre 'a'. Compter le nombre de tels mots (avec le programme) et essayer de deviner une formule pour ce nombre.
4. (**) Faire une fonction qui prend en arguments deux entiers n et k et qui affiche tous les mots de longueur n utilisant les k premières lettres de l'alphabet.

Exercice 6 : () Permutation d'un ensemble à n elements**

Une permutation d'un ensemble à n éléments peut être uniquement représentée par une suite contenant chaque entier de 1 à n une et une seule fois. En C on pourra représenter une permutation par un tableau de taille n contenant des entiers de 1 à n . Chaque élément étant présent exactement une fois dans le tableau. Ecrire un programme qui affiche toutes les permutations possibles à n éléments.