Feuille de TD n°2

Exercice 1. *Écrire une fonction qui prend un nombre ≥ 1 en entrée et qui affiche un triangle à l'aide du caractère « * ». Par exemple pyramide(3) devrait afficher :*

```
*
**
***
**
*
```

Exercice 2. *Même question mais cette fois la pyramide devrait être à l'horizontale et pyramide(3) devrait afficher :*

```
  *
 * *
* * *
```

Exercice 3. *Écrire une fonction prenant en entrée un nombre naturel strictement positif et retournant un booléen suivant si l'entier est premier ou non.*

Exercice 4. *La conjecture de Goldbach affirme que tout nombre pair supérieur à trois est la somme de deux nombres premiers. Écrire un algorithme pour vérifier cette conjecture jusqu'à un rang N passé en paramètre. La fonction prendra un entier N en paramètre et retournera un booléen.*

Exercice 5. *Écrire une fonction qui prend en entrée deux listes $d = [h, m, s]$ et $d' = [h', m', s']$ représentant des temps en heures, minutes, secondes (trois entiers) et qui retourne le temps $d' - d$ (supposé positif) sous forme d'une liste $[h'', m'', s'']$.*

Exercice 6. *On dispose d'une fonction qui donne le lendemain d'un jour (de lundi à dimanche), que l'on note :* `jourSuivant : jour ↦ jourSuivant(jour).` *On se propose de déterminer quel jour nous serons dans N jours, et nous notons cette fonction :*

`jourLointain : (jour,N) ↦ jourLointain(jour,N).`

- *Proposer un algorithme qui donne le "jour lointain" de manière itérative.*
- *Proposer un algorithme qui donne le "jour lointain" de manière récursive.*

Exercice 7.

1. *Proposer un algorithme itératif, qui prend en entrée $p, n \in \mathbb{N}$, et qui renvoie $\binom{n}{p}$).*
2. *Soient $p, n \in \mathbb{N}$. Que vaut $\binom{n}{p}$ lorsque $p > n$? Lorsque $p = n$? Rappeler la formule du triangle de Pascal. Proposer un algorithme récursif qui prend en entrée $p, n \in \mathbb{N}$ et qui renvoie $\binom{n}{p}$.*

Exercice 8. *On suppose que l'on dispose d'un système qui sait effectuer des additions, mais pas des multiplications. On peut par contre utiliser des boucles et des appels récursifs. On se pose alors le problème suivant : comment calculer le produit pq lorsque p et q sont deux entiers positifs ?*

Écrire deux algorithmes, l'un itératif, l'autre récursif, qui calculent le produit pq.

Peut-on réduire le nombre d'opérations ou d'appels récursifs en comparant p et q ?