Feuille de TD n°1

Exercice 1. *Écrire un algorithme qui, prenant en entrée un réel a et un entier positif n, calcule le produit de ces deux valeurs en n'utilisant que l'addition.*

Exercice 2.
1. *Écrire un programme qui prend en entrée trois réels a, b et c et qui renvoie le minimum de ces trois nombres.*
2. *Écrire un programme qui prend en entrée une liste de nombre réels L, et qui renvoie le minimum de cette liste.*
3. *Modifier le programme précédent, pour qu'il renvoie le minimum m de la liste, ainsi qu'un élément i tel que $L[i] = m$.*

Exercice 3. *Soient $\{u_i\}_{1\leq i\leq N}$ des valeurs réelles. Soit $L = [u_1, \dots, u_N]$.*
1. *Proposer un algorithme qui permette de calculer $S_N = \sum_{i=1}^{N} u_i$.*
2. *Proposer un algorithme qui permette de calculer $S_N = \prod_{i=1}^{N} u_i$.*
3. *Proposer un algorithme pour compter le nombre d'éléments positifs parmi les u_i.*
4. *Écrire un algorithme qui permette de calculer la moyenne de tous les éléments, puis la moyenne des éléments positifs, et la moyenne des éléments négatifs. On pourra utiliser les algorithmes précédents, lorsque c'est pertinent.*

Exercice 4.
1. *Soit une fonction $f : \mathbb{R} \to \mathbb{R}$ et $(x_n) \in \mathbb{R}^{\mathbb{N}}$ définie par :*
$$\begin{cases} x_0 \in \mathbb{R} \\ x_{n+1} = f(x_n), \quad \forall n \geq 0. \end{cases}$$
Proposer deux algorithmes, l'un itératif, l'autre récursif, qui calculent x_N pour N donné.
2. *A présent, on se donne $g : \mathbb{R}^2 \mapsto \mathbb{R}$ et on définit $(x_n) \in (\mathbb{R}^2)^{\mathbb{N}}$ par :*
$$\begin{cases} x_0 \in \mathbb{R} \\ x_1 \in \mathbb{R} \\ x_{n+1} = g(x_n, x_{n-1}), \quad \forall n \geq 1. \end{cases}$$
De nouveau, proposer un algorithme qui calcule x_N pour N donné.

Exercice 5.
1. *Écrire un algorithme qui prend en entrée une liste $[a_k, \dots, a_0]$ et qui renvoie $\sum_{i=0}^{k} a_i 2^i$.*
2. *Écrire 25, 65 et 19 en base 2.*
3. *Écrire un algorithme qui prend en entrée un entier naturel n et qui renvoie la liste $[a_k, \dots, a_0]$ d'entiers entre 0 et 1 telle que $n = \sum_{i=0}^{k} a_i 2^i$ et $a_k \neq 0$.*
4. *Écrire $36, 11$ et 71 en base 3.*
5. *Écrire un algorithme qui prend en entrée un entier naturel n et qui renvoie la liste $[a_k, \dots, a_0]$ d'entiers entre 0 et 2 telle que $n = \sum_{i=0}^{k} a_i 3^i$ et $a_k \neq 0$.*

Exercice 6. *Soit n un entier strictement positif.*
- *On appelle **diviseur propre** de n un diviseur de n distinct de n.*
- *n est dit **parfait** s'il vaut la somme de ses diviseurs propres.*
- *n est dit **chanceux** si $(n + m + m^2)$ est premier pour tout $0 \leqslant m < n - 1$.*

1. *Écrire un algorithme Div(n) qui, prenant en entrée un entier strictement positif n, calcule la somme de ses diviseurs propres.*
2. *Écrire un algorithme Parfait(n) qui renvoie vrai si l'entier strictement positif n pris en entrée est parfait et faux sinon.*
3. *Idem pour l'algorithme Chanceux(n) en supposant qu'on dispose d'un test de primalité Premier(m) qui renvoie vrai si m est premier et faux sinon.*

Exercice 7. *On considère une suite récurrente (u_n) donnée par $u_0 = 1$, $u_1 = 3$, et les formules de récurrence :*
$$\begin{cases} u_{2n} = (u_n)^2 + 5 \\ u_{2n+1} = u_n u_{n+1} + 7 \end{cases}, \text{ si } n \in \mathbb{N}^*.$$

1. *Recopier et compléter l'algorithme suivant, de sorte que* **suite**(n) *renvoie le couple* (u_n, u_{n+1}).

   suite(n)
   si $n = 0$:
   | *Renvoyer* $(1, 3)$
   si $n = 1$:
   | *Renvoyer* $(?, ?)$
   $k \leftarrow n//2$
   $(a, b) \leftarrow$ **suite**$(?)$
   si $n\%2 = 0$
   | *Renvoyer* $(?, ?)$
   sinon
   | *Renvoyer* $(?, ?)$

   On justifiera la validité de l'algorithme obtenu.
2. *Soit* $n \in \mathbb{N}^*$. *Montrer qu'il existe un unique* $\ell \in \mathbb{N}$ *tel que* $2^\ell \leq n < 2^{\ell+1}$. *Exprimer* ℓ *à l'aide de* $\log_2(n)$.
3. *Pour tout entier* $n \geq 2$, *exprimer en fonction de* n *le nombre d'appels récursifs que réalise* **suite**(n).

Exercice 8. *Soit la suite de Fibonacci, définie par :*

$$\begin{cases} u_0 &= 0 \\ u_1 &= 1 \\ u_n &= u_{n-1} + u_{n-2}, \quad \forall n \geq 2. \end{cases}$$

1. *Proposer un algorithme qui prend en entrée un entier* N *et qui renvoie* u_N.
2. *Proposer deux algorithmes, l'un itératif, l'autre récursif, qui prennent en entrée un entier* N *et qui renvoient le couple* (u_N, u_{N+1}).
3. *Soit* $L \geq 1$ *un entier. Modifier l'algorithme proposé à la première question afin de répondre à la question suivante : «Quel est le premier terme de la suite de Fibonacci (et son rang) tel que* $u_n \geq L$ *?» On justifiera la validité de l'algorithme avec soin.*

Remarque. *On peut aussi calculer* (u_n) *en utilisant la formule suivante :* $u_n = \frac{1}{2^n \sqrt{5}} \left[(1+\sqrt{5})^n - (1-\sqrt{5})^n\right]$.

Exercice 9.

1. *Écrire un algorithme, que l'on notera DivEuc(a,b), qui à partir de deux entiers* $(a, b) \in \mathbb{N} \times \mathbb{N}^*$, *donne le quotient et le reste de la division euclidienne de* a *par* b.
2. *Pour* $(a, b) \in \mathbb{N} \times \mathbb{N}^*$, *on a l'algorithme suivant :*

   DE(a,b)
   $Q \leftarrow 0$
   $R \leftarrow a$
   tant que $R \geq b$
   | $n \leftarrow 0$
   | $c \leftarrow 1$
   | *tant que* $10^n b \leq R$
   | | $n \leftarrow n + 1$
   | $n \leftarrow n - 1$
   | *tant que* $10^n bc \leq R$
   | | $c \leftarrow c + 1$
   | $c \leftarrow c - 1$
   | $Q \leftarrow Q + 10^n c$
   | $R \leftarrow R - 10^n bc$
   renvoyer (Q, R)

   Quel résultat renvoie cet algorithme ? Justifier.
   Indication : Montrer qu'à chaque étape de la boucle indicée par k *dans l'algorithme, la relation* $a = bQ_k + R_k$ *est vraie (ce qu'on appelle un invariant de boucle).*