Corrigé

Exercice 1 :

1)

Algorithme Valeur_absolue(x)
si $x < 0$:
  $|x \leftarrow -x$
Renvoyer x.

2)

Algorithme Factorielle(n)
$a \leftarrow 1$
pour i de 1 à n :
  $|\ a \leftarrow ai$
Renvoyer a.

Cet algorithme renvoie $n!$ car si a_i désigne la valeur de a à la fin de l'étape i, on a $a_i = i!$ pour tout $i \in [\![1, n]\!]$.

3)

Algorithme Puissance(n, x)
$a \leftarrow 1$
pour i de 1 à n :
  $|\ a \leftarrow ax$
Renvoyer a.

Algorithme Puissance_rec(n, x)
si $n = 0$:
  | Renvoyer 1 :
sinon :
  | Renvoyer xPuissance_rec($n - 1, x$)

4)

Algorithme Somme_serie(n, x)
$a \leftarrow 0$
pour i de 0 à n :
  | $a \leftarrow a+$Puissance(i, x)/Factorielle(i)
Renvoyer a.

Algorithme Somme_serie2(n, x)
$a \leftarrow 0$
$y \leftarrow 1$
pour i de 1 à n :
  $|\ y \leftarrow y.\frac{x}{i}$
  $|\ a \leftarrow a + y$
Renvoyer a.

Pour $i \in [\![1, n]\!]$, si y_i et a_i désignent les valeurs de y et a à la fin de la boucle, on a $y_i = \frac{x^i}{i!}$ et $a_i = \sum_{j=1}^{i} y_j = \sum_{j=1}^{i} \frac{x^j}{j!}$.

5)

Algorithme Maximum(L)
$n \leftarrow$ long(L)
$M \leftarrow L[0]$
pour i allant de 1 à $n-1$:
| si $L[i] > M$:
| $M \leftarrow L[i]$
Renvoyer M.

On note M_i la valeur de M à la fin de l'étape i. Alors $M_1 = \max(L[0], L[1])$. Soit $i \in [\![1, n-1]\!]$ et supposons que $M_i = \max(L[0], \ldots, L[i])$. Alors $M_{i+1} = \max(M_i, L[i+1]) = \max(L[0], \ldots, L[i+1])$. Par récurrence, on en déduit que $M_{n-1} = \max(L[0], \ldots, L[n-1]) = \max(L)$.

Exercice ??

1)

Algorithme Majorant(M)
$S \leftarrow 1$
$n \leftarrow 1$
tant que $S < M$:
| $n \leftarrow n+1$
| $S \leftarrow S + \frac{1}{n}$
Renvoyer n.

Exercice 2

1)

suite_u(n) :
$a \leftarrow 1$ # a correspond à u_n
$b \leftarrow 2$ # b correspond à u_{n+1}
Pour i allant de 1 à $n-1$:
| $a, b \leftarrow b, 3a + 2b$
Si $n = 0$:
|Renvoyer a
Sinon :
|Renvoyer b.

2)

suite_u_rec(n) :
Si $n <= 1$:
|Renvoyer $n+1$
Sinon :
|Renvoyer $3 * suite\_u\_rec(n-2) + 2 * suite\_u\_rec(n-1)$.

3)

suite_v(n) :
Si $n <= 1$:
|Renvoyer $1 + 2n$
sinon si $n\%2 = 0$:
|Renvoyer $(\text{suite\_v}(n/2))^2 + 5$
sinon :
|Renvoyer $(\text{suite\_v}((n-1)/2)).(\text{suite\_v}((n-1)/2+1)) + 7$.

Exercice 3 : cf cours.

Exercice 4

Dans le programme suivant, Ent désigne la fonction partie entière inférieure.

solutions() :

$L \leftarrow [\,]$
Pour y de 1 à 100
| Si Ent($\sqrt{1+2y^2}$) $= \sqrt{1+2y^2}$:
| $L \leftarrow L + [(\sqrt{1+2y^2}, y)]$
Renvoyer L.

Exercice 5

La suite (u_n) est strictement croissante car $u_{n+1} - u_n > 0$ pour tout $n \in \mathbb{N}^*$. Soit $n \in \mathbb{N}^*$. Alors $v_{n+1} - v_n = \frac{1}{(n+1)^2 n!} - \frac{1}{n.n!} = \frac{1}{n!}(\frac{1}{(n+1)^2} - \frac{1}{n}) < 0$. La suite (v_n) est donc strictement décroissante. Comme $v - u$ tend vers 0, les suites u et v sont adjacentes.

Pour tout $n \in \mathbb{N}$, on a $u_n < e < v_n = u_n + \frac{1}{n.n!}$, donc $0 < e - u_n < \frac{1}{n.n!}$. On peut donc utiliser le programme suivant :

approximation_e(ϵ) :

$s \leftarrow 2$
$P \leftarrow 1$
$n \leftarrow 1$
Tant que $1/(P * n) > \epsilon$:
$|n \leftarrow n+1$
$|P \leftarrow P * n$
$|s \leftarrow s + 1/P$
Renvoyer s.

Exercice 6

1) Cet algorithme ne fonctionne pas. En effet, à la première ligne du « tant que », b devient égal à a, donc on obtient $b = 0$ à la deuxième ligne du « tant que ». On peut corriger le problème en rajoutant une variable intermédiaire.

Algorithme(a, b) :

si $a <= b$:
| $b, a \leftarrow a, b$
Tant que $r \neq 0$:
| $r \leftarrow a\%b$
$|a \leftarrow b$
$b \leftarrow r$
Renvoyer a.

2) Cet algorithme ne fonctionne pas. En effet, à chaque fois que random() est utilisée elle produit un nouveau nombre pseudo-aléatoire. Par exemple il est possible que pour un t donné, aucune des conditions imposées dans les « si » ne soit vérifiée. On peut le corriger de la façon suivante :

Algorithme :

$x = 0$
Pour t allant de 1 à 50 faire :
$r \leftarrow$ `random`()
| Si $r < 1/3$:
| $x \leftarrow x + 1$
| Si $r >= 1/3$ et $r < 2/3$:
| $x \leftarrow x + 2$
|Si $r > 2/3$
| $x \leftarrow x - 3$
Renvoyer x.