L'Algorithmique

1. Les bases

Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné.
Pour fonctionner, un algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter.
Les ordinateurs, les calculatrices, ne sont capables de comprendre que quatre catégories d'instructions.
Ces quatre familles d'instructions sont :

- l'affectation de variables
- la lecture / écriture
- les tests
- les boucles

Un algorithme informatique se ramène à la combinaison de ces quatre instructions de base.
L'écriture de l'algorithmique est indépendant des particularités de tel ou tel langage.
Pour écrire un algorithme on utilise généralement une série de conventions appelée « pseudo-code ».
Ce pseudo-code est susceptible de varier légèrement d'un livre (ou d'un enseignant) à un autre.
Aucune machine n'est censée le reconnaître.

2. Les variables

Pour mémoriser les données initiales, ou les résultats intermédiaires des calculs, on utilise des "variables".
Du point de vue algorithmique, une variable a un nom fixe et une valeur qui peut changer au cours de l'exécution de l'algorithme.

a) Entrées Sorties

Pour obtenir une donnée entrée au clavier, on utilise l'instruction : *Lire la valeur de a*
Cette instruction, non seulement lit une valeur tapée au clavier, mais affecte aussi cette valeur à la variable *a*.
Pour afficher un message et un résultat, on utilise : *Afficher « le résultat est : » b*

b) Variables, affectations et manipulation des données

Après chaque étape, pour chacun des algorithmes, décrire l'état des variables a, b ou c après avoir rentré les données si besoin.

Entrées Lire la valeur de a Lire la valeur de b **Traitement** Donner à b la valeur de a Donner à a la valeur de b **Sortie** Afficher a et b	**Initialisations** Donner à a la valeur 1 Donner à b la valeur 2 Donner à c la valeur 3 **Traitement** $c := a$ $a := b$ $b := c$ **Sortie** Afficher b	Début Lire la valeur de a Donner à b la valeur de a^2 Donner à b la valeur de $2b$ Donner à b la valeur $b - 5a$ Donner à b la valeur $b + 3$ Afficher b Fin

3. Les structures

a) Structures répétitives

Il s'agit de répéter un bloc d'instructions plusieurs fois de suite. Les deux variantes principales consistent à :

- répéter un bloc d'instructions un nombre de fois donné : **boucle POUR** ;
- répéter un bloc d'instructions jusqu'à ce qu'une condition soit vérifiée (ou tant qu'une condition est vérifiée) : **boucle TANT QUE**.

Exemples :

Faire fonctionner les 2 algorithmes suivants (décrire à chaque étape l'état des variables utilisées). Que font-ils ?

Initialisations
Donner à *fact* la valeur 1
Traitement
Pour i de 1 jusqu'à 10 faire
    Donner à *fact* la valeur $fact \times i$
FinPour
Sortie
Afficher *fact*
	Début
	  Donner à *fact* la valeur 1
	  Donner à i la valeur 1
	  Tant que $i \leq 10$ faire
	      Donner à *fact* la valeur $fact \times i$
	      Donner à i la valeur $i + 1$
	  Fin Tantque
	  Afficher *fact*
	Fin

Exercice : Faire fonctionner les 4 algorithmes suivants (décrire à chaque étape l'état des variables utilisées).
Qu'en pensez-vous ?

Début Donner à *fact* la valeur 1 Donner à i la valeur 1 Tant que $i \leq 10$ faire |Donner à *fact* la valeur *fact*$\times i$ Fin Afficher *fact* Fin	Début Donner à *fact* la valeur 1 Tant que $i \leq 10$ faire Donner à *fact* la valeur *fact*$\times i$ Donner à i la valeur $i + 1$ Fin Afficher *fact* Fin

Début Donner à *fact* la valeur 1 Donner à i la valeur 1 Tant que $i \leq 10$ faire Donner à i la valeur 1 Donner à *fact* la valeur *fact* $\times i$ Donner à i la valeur $i + 1$ Fin Afficher *fact* Fin	Début Donner à *fact* la valeur 1 Donner à i la valeur 11 Tant que $i \leq 10$ faire Donner à *fact* la valeur *fact* $\times i$ Donner à i la valeur $i + 1$ Fin Afficher *fact* Fin

b) **Structures alternatives**

Ce sont des instructions de contrôle, pour exécuter des instructions seulement dans le cas où une condition est réalisée et d'autres dans le cas où elle ne l'est pas.

Exemple

Faire fonctionner l'algorithme ci-contre, que fait-il ?

Entrées
Lire le nombre a
Lire le nombre b
Traitement/Sorties
Si $a < b$ alors
Afficher a
Afficher b
Sinon
Afficher b
Afficher a
FinSi

Exercice : Ecrire un algorithme qui demande 3 paramètres a, b, c coefficients d'un polynôme de second degré.
Le programme devra ensuite donner le nombre et la valeur de ses racines réelles

c) **Algorithme mystère**

Le faire fonctionner pour $a = 5$ et $n = 7$. Que fait cet algorithme ?

Début
Lire la valeur du réel a et la valeur de l'entier n
Donner à la variable *res* la valeur 1
Donner à la variable *puiss* la valeur n
Donner à la variable *temp* la valeur a
Tant que *puiss* est non nulle faire
Si *puiss* est impaire alors
Donner à *res* la valeur *temp* $\times$ *res*
Donner à *puiss* la valeur *puiss* $- 1$
Fin
Donner à *puiss* la valeur *puiss* $\div 2$
Donner à *temp* la valeur *temp* $\times$ *temp*
Fin
Afficher *res*
Fin

4. Exemples d'algorithmes

a) Exemple 1

1. Que fait l'algorithme ci-contre ?

2. Ecrire un algorithme qui calcule la somme des carrés des n premiers entiers non nuls

```
Entrée
Lire la valeur de l'entier naturel n
Initialisation
Donner à S la valeur 0
Traitement
Pour k de 1 jusqu'à n faire
  | Donner à S la valeur  S + k
FinPour
Sortie
Afficher S
```

b) Exemple 2 (dichotomie pour résoudre l'équation $f(x) = 0$).

f désigne une fonction continue et strictement monotone sur un intervalle $[a ; b]$.

1. Que retourne l'algorithme suivant à la fin de son exécution ?

2. Expliquer ce que représente e.

```
Entrées
Lire la fonction f
Lire les valeurs réelles  a, b, e
Traitement
Si f(a) × f(b) ≥ 0  alors
  | Afficher « 0 sol entre a et b »
  | Sinon
  | Tant que b – a > e  faire
  |   | Donner à c  la valeur (a + b)/2
  |   | Si  f(a) × f(c) ≤ 0   alors
  |   |   | Donner la valeur c  à b
  |   |   | Sinon
  |   |   | Donner la valeur c  à a
  |   | FinSi
  | FinTantque
FinSi
Sorties
Afficher a et b
```

c) Exemple 3

On considère la suite (u_n) définie par : $u_0 = 1$ et $u_{n+1} = u_n - \ln(u_n^2 + 1)$

Que retourne l'algorithme suivant à la fin de son exécution ?
Quel résultat mathématique permet d'affirmer que cet algorithme se termine ?

```
Début
  | Donner à U la valeur 1
  | Donner à k  la valeur 0
  | Tant que U > 0,05  faire
  |   | Donner à k la valeur k + 1
  |   | Donner à U la valeur   U – ln(U^2 + 1)
  | Fin
  | Afficher k
Fin
```

d) Exemple 4

On considère la suite (u_n) définie par

$$u_0 = \frac{1}{3} \text{ et pour } n \geq 0, \quad u_{n+1} = \frac{3}{2}u_n - \frac{1}{3}$$

1. Démontrer que pour tout entier naturel n, $u_n \leq \dfrac{2}{3}$

2. On considère l'algorithme suivant :

   a) Que fait cet algorithme ?
   b) Justifier qu'il ne se termine pas

3. Ecrire un algorithme qui calcule u_{20}

```
Début
  | Donner à U la valeur 1/3
  | Donner à k la valeur 0
  | Tant que U < 0,9   faire
  |   | k = k + 1
  |   | Donner à U la valeur (3/2)U – 1/3
  | Fin
  Afficher k
  Afficher U
Fin
```

4. L'algorithme suivant voudrait calculer la somme des 20 premiers termes, mais deux erreurs se sont glissées à l'intérieur, le rectifier pour qu'il soit correct.

Initialisations

$U := \frac{1}{3}$

$S := 0$

Traitement

Pour k de 0 jusqu'à 20 faire

  $U := \frac{3}{2}U - \frac{1}{3}$

  $S := S + U$

FinPour

Sortie

Afficher S

e) Exemple 5

On considère les suites (a_n) et (b_n) définies $a_0 = 1$ et $b_0 = 7$ et pour tout entier naturel n, $\begin{cases} a_{n+1} = \dfrac{2a_n + b_n}{2} \\ b_{n+1} = \dfrac{a_n + 2b_n}{3} \end{cases}$

Des 3 algorithmes suivants, un seul est exact, lequel ? Que fait-il ?

(Pour simplifier les écritures on emploie aussi cette notation $a \leftarrow 2$ pour donner la valeur 2 à la variable a)

Début Lire la valeur de l'entier n $a \leftarrow 1$ $b \leftarrow 7$ $k \leftarrow 0$ Tant que $k < n$ faire $a \leftarrow \frac{2a+b}{2}$ $b \leftarrow \frac{a+2b}{3}$ $k \leftarrow k+1$ Afficher a , b Fin	Début Lire la valeur de l'entier n $a \leftarrow 1$ $b \leftarrow 7$ $k \leftarrow 0$ Tant que $k < n$ faire $a \leftarrow \frac{2a+b}{2}$ $temp \leftarrow a$ $b \leftarrow \frac{a+2b}{3}$ $k \leftarrow k+1$ Afficher a , b Fin	Début Lire la valeur de l'entier n $a \leftarrow 1$ $b \leftarrow 7$ $k \leftarrow 0$ Tant que $k < n$ faire $temp \leftarrow a$ $a \leftarrow \frac{2temp+b}{2}$ $b \leftarrow \frac{temp+2b}{3}$ $k \leftarrow k+1$ Afficher a , b Fin