Algorithmique – Premiers pas

I. Définitions :

Algorithme :
Un algorithme est une suite d'instructions donnée pour résoudre un problème. Ces instructions concernent des variables qui peuvent être des nombres, des listes (de nombres) ou des chaînes de caractères (par exemple des mots ou des codes de cartes bleues).
Il y a quatre types d'instruction :
1°) L'affectation de variables :
Exemples :
a) X `prend la valeur` 3 (si avant cette instruction X valait 8, après cette instruction X vaut …….)
b) X `prend la valeur` 5+X (si avant cette instruction X valait 8, après cette instruction X vaut ……..)
2°) Lecture écriture :
Exemples :
a) `Lire` X (cette instruction signifie qu'on demande à l'utilisateur de l'algorithme de choisir une valeur pour la variable X)
b) `Afficher` X (signifie que l'algorithme fait apparaître la valeur actuelle de la variable X)
c) `Afficher` « l'algorithme est terminé » (cette instruction fait apparaitre le message entre guillemets)
3°) Les tests :
Exemples :
a) `Si` X < 5 `alors` X `prend la valeur` 3 (si avant cette instruction X valait 8, après cette instruction X vaut …….. , mais si avant cette instruction X valait 4, après cette instruction X vaut ……..)
b) `Si` X > 10 `alors` X `prend la valeur` 2 `sinon` X `prend la valeur` 1(si avant cette instruction X valait 8, après cette instruction X vaut ……., mais si avant cette instruction X valait 14, après cette instruction X vaut …….)
4°) Les boucles :
Exemples :
a) `Pour` *i* `allant de` 1 `à` 3 : X `prend la valeur` 5+X (si avant cette instruction X valait 9, alors X va prendre successivement les valeurs.....)
b) `Tant que` X < 40 : X `prend la valeur` 5+X (si avant cette instruction X valait 12, alors X va prendre successivement les valeurs …….....)
Remarque 1 :
Il y a une grande différence entre ces deux types de boucles, car avec la boucle `Pour…` on sait que de toutes façons la séquence d'instruction sera répétée trois fois quelque soit la valeur de la variable X, alors qu'avec la boucle `Tant que…` si on ne connaît pas la valeur de la variable X, on ne peut pas dire combien de fois la séquence d'instruction sera répétée, ni même si elle sera effectuée au moins une fois.
Remarque 2 :
Un algorithme peut être écrit dans un « langage naturel », c'est-à-dire compréhensible par n'importe quelle personne, ou bien être écrit dans un langage de programmation qui sera compris par un ordinateur.

II. Exemples :

Premier exemple :
On considère l'algorithme suivant :

Entrée
A est un nombre réel
B est un nombre réel
C est un nombre réel
Traitement
M `prend la valeur` (A+B+C)/3
Sortie
`Afficher` M

1. Combien y a-t-il de variables dans cet algorithme ?
2. Quelles sont les variables non affichées ?

3. Si on donne aux variables A, B et C les valeurs respectives 4 ; 6 et 11, quelle sera la valeur affichée pour M ?
4. Quel est le rôle de cet algorithme ?

Deuxième exemple :

On considère l'algorithme suivant :

```
Entrée
A est un nombre réel
B est un nombre réel
C est un nombre réel
D est un nombre réel
Traitement
m prend la valeur A
M prend la valeur A
Si B < m alors m prend la valeur B
Si C < m alors m prend la valeur C
Si D < m alors m prend la valeur D
Si B > M alors M prend la valeur B
Si C > M alors M prend la valeur C
Si D > M alors M prend la valeur D
Sortie
Afficher m
Afficher M
```

1. Combien y a-t-il de variables dans cet algorithme ?
2. Quelles sont les variables non affichées ?
3. On donne aux variables A, B, C et D les valeurs respectives 4 ; 6 ; 3 et 11, Compléter le tableau suivant :

	m	M
m prend la valeur A M prend la valeur A		
Si B < m alors m prend la valeur B		
Si C < m alors m prend la valeur C		
Si D < m alors m prend la valeur D		
Si B > M alors M prend la valeur B		
Si C > M alors M prend la valeur C		
Si D > M alors M prend la valeur D		

Quelle valeur affiche M ?

Quelle valeur affiche m ?

5. On donne aux variables A, B, C et D les valeurs respectives 15 ; 17 ; 11 et 16, Compléter le tableau suivant :

	m	M
m prend la valeur A M prend la valeur A		
Si B < m alors m prend la valeur B		
Si C < m alors m prend la valeur C		
Si D < m alors m prend la valeur D		
Si B > M alors M prend la valeur B		
Si C > M alors M prend la valeur C		
Si D > M alors M prend la valeur D		

Quelle valeur affiche M ?

Quelle valeur affiche m ?

6. Quel est le rôle de cet algorithme ?

Troisième exemple :
Dans le but de simuler une expérience aléatoire, on dispose d'un algorithme écrit en langage naturel :

```
Initialisation :
S prend la valeur 0
Traitement
Pour I allant de 1 jusqu'à 3
    | F prend la valeur d'un entier aléatoire de 1 à 6
    | S prend la valeur de F+S
Fin pour
Sortie
Afficher S
```

1. Combien y a-t-il de variables dans cet algorithme ?
2. Quelles sont les variables non affichées ?
3. Dans cet algorithme, des instructions sont répétées. Lesquelles ? Combien de fois ?
4. Compléter le tableau suivant qui donne un exemple d'exécution de l'algorithme :
   Initialisation : S=0

I=1	F=	S=
I=2	F=	S=
I=3	F=	S=

   Quelle valeur affiche S ?

5. Quelle est la plus petite valeur que S peut prendre ? La plus grande ?

6. Quelle expérience cet algorithme permet-il de simuler ?

7. Calculer la probabilité pour que S prenne la valeur 3.

Quatrième exemple :
On considère l'algorithme suivant :

```
Entrée
N est un entier naturel
Initialisation :
P prend la valeur 0
U prend la valeur 4
S prend la valeur 4
Traitement
Tant que P < N
    | P prend la valeur P+1
    | U prend la valeur 4+2P
    | S prend la valeur S+U
Sortie
Afficher S
```

Compléter le tableau suivant pour N=5 :
Initialisation :S=0

	P	U	S
Initialisation			
Etape 1			

Etape 2			

Quelle valeur affiche S ?

Cinquième exemple :

On cherche un algorithme qui range deux nombres dans l'ordre croissant. On propose un premier algorithme :

```
Entrée
a est un nombre réel
b est un nombre réel
Traitement
Si b < a alors
    | b prend la valeur a
    | a prend la valeur b
Sortie
Afficher a
Afficher b
```

Pourquoi cet algorithme est-il incorrect ? Donner un cas où il fonctionne et un cas où il ne fonctionne pas.

On propose un deuxième algorithme qui lui est correct :

```
Entrée
a est un nombre réel
b est un nombre réel
Traitement
Si b < a alors
    | Échanger a et b
Sortie
Afficher a
Afficher b
```

On voit que le problème est d'échanger les valeurs de a et b, or cette instruction n'est pas directement disponible dans certains langages. Donner une solution utilisant une troisième variable :

```
Entrée
a est un nombre réel
b est un nombre réel
Traitement
Si b < a alors
    |
    |
    |
Sortie
Afficher a
Afficher b
```