

## Algorithmique et Suites numériques

### Utiliser un algorithme avec les suites

#### 1°) Écrire, analyser et modifier un algorithme

**a) Écrire** un algorithme pour que, si on saisit une valeur de  $N$ , il permet de calculer le terme de rang  $N$  d'**une suite explicite** définie par sa fonction associée  $u_n = f(n)$  et d'afficher le résultat.

#### ALGO.01

<b>Entrée</b>	
$N$ un nombre entier	
$U$ un nombre réel	
<b>Initialisation</b>	
<b>Traitement</b>	
Lire $N$	<i>(Permet de « saisir » <math>N</math> par l'utilisateur)</i>
Affecter à $U$ la valeur $f(N)$	<i>(Remplacer <math>f(N)</math> par l'expression en fonction de <math>N</math>.)</i>
Afficher Message « $U($ »	<i>(Permet d'« afficher » le texte <math>U(N) = \text{résultat}</math>)</i>
Afficher $N$	
Afficher Message « $) =$ »	
<b>Sortie</b>	
Afficher $U$	<i>(Permet d'« afficher » la dernière valeur de <math>U</math>.)</i>

**b) « Analyser »** ce que fait un algorithme, revient à construire un «tableau d'étapes »

**Définition** : Un **tableau d'étapes** contient en première ligne toutes les *variables d'entrée et de sortie* et dans les lignes suivantes toutes les étapes d'exécution (dans l'ordre) des instructions de l'algorithme.

Dans l'exécution suivante de l'algorithme, l'utilisateur choisit  $N = 5$ , l'algorithme va afficher :  $U(5) = \text{valeur de } U(5)$ .

#### TAB.01

Variables	$N$	$U$	
Initialisation	- (inutile ici)	- (inutile ici)	
Étape n°1	5	$U(5)$	

Ici, les calculs sont immédiats.

c) On peut « **modifier** » cet algorithme pour que, si on saisit une valeur de  $N$ , il affiche tous les termes de  $U(0), \dots$  à  $U(N)$ . Autrement dit, on obtient un tableau de valeurs :

### ALGO.02

#### Entrée

$N$  un nombre entier  
 $k$  un nombre entier  
 $U$  un nombre réel

#### Initialisation

#### Traitement

Lire  $N$  (Permet de « **saisir** »  $N$  par l'utilisateur)  
 Pour  $k$  allant de 0 à  $N$  (Boucle de **calcul terme-à-terme** jusqu'à  $U_N$ .)  
     Debut\_de\_Pour  
     Affecter à  $U$  la valeur  $f(k)$  (Remplacer  $f(N)$  par **son expression**.)  
     Afficher Message «  $U(k)$  » (Pour pouvoir afficher  $U(k) = \text{résultat}$ )  
     Afficher  $k$   
     Afficher Message «  $) =$  »  
     Afficher  $U$  Cocher « Retour à la ligne [x] »  
     Fin\_de\_Pour

#### Sortie

Ici, on affiche plusieurs valeurs et l'affichage se fait « dans la boucle ».

Dans l'exécution suivante de l'algorithme, l'utilisateur choisit  $N = 5$ , l'algorithme va afficher :  $U(0) = \text{valeur de } U(0)$ , puis  $U(1) = \text{valeur de } U(1), \dots$  jusqu'à  $U(5) = \text{valeur de } U(5)$ .

### TAB.02

Variables	$N$	$k$	$U$
Initialisation	- à saisir	0	- (inutile ici)
Étape n°1	5	-	-
Étape n°2	5	0	$U(0)$
Étape n°3	5	1	$U(1)$
Étape n°4	5	2	$U(2)$
Étape n°5	5	3	$U(3)$
Étape n°6	5	4	$U(4)$
Étape n°7	5	5	$U(5)$

Ici, les calculs sont immédiats.

Attention :

Si l'instruction « **Afficher**  $U$  » se trouve à l'extérieur de la boucle « **Pour** » l'algorithme affichera uniquement la dernière valeur de  $U$ , soit  $U(5) = \text{valeur de } U(5)$ .

b) Calcul du terme de rang N d'**une suite récurrente** de premier terme  $u_0$  et  $u_{n+1}=g(u_n)$ .

### Entrée

$k$  un nombre entier

$N$  un nombre entier

$U$  un nombre réel

### Initialisation

Affecter à  $U$  la valeur  $U_0$  permet d'« **initialiser** » la valeur de  $U$  ( $u_0$  donnée)

### Traitement

Lire  $N$  permet de « **saisir** »  $N$  par l'utilisateur

Pour  $k$  allant de 1 à  $N$  Boucle de **calcul terme-à-terme** jusqu'à  $U_N$ .

Debut\_de\_Pour

Affecter à  $U$  la valeur  $g(U)$  (Remplacer  $g(U)$  par son expression.)

Afficher Message «  $U$  » (Pour pouvoir afficher  $U(N) = \text{résultat}$ )

Afficher  $k$

Afficher Message «  $) =$  »

Afficher  $U$

Fin\_de\_Pour

### Sortie

Ici, on affiche plusieurs valeurs et l'affichage se fait « dans la boucle ».

2°) Trouver le plus petit entier vérifiant une condition.

**Exemple 1** : On sait que la suite définie explicitement par  $u_n = 5 \times \left(\frac{3}{5}\right)^n$  est une suite géométrique convergente et tend vers 0. Écrire un algorithme tel que :

Si  $M$  est un nombre positif donné, trouver le plus petit entier  $N_0$  tel que  $U_{N_0} < M$

### Entrée

$k$  un nombre entier

$U$  un nombre réel

$M$  un nombre réel

### Initialisation

Affecter à  $k$  la valeur 0 permet d'« **initialiser** » la valeur de  $k$ .

### Traitement

Lire  $M$  permet de « **saisir** »  $M$  par l'utilisateur.

Tant\_que ( $U \geq M$ ) Faire continuer la boucle de Test jusqu'à obtenir  $U < M$ ,

Debut de Tant\_que puis sortir de la boucle

Affecter à  $k$  la valeur  $k+1$  permet d'**incrémenter**  $k$  (ie **augmenter  $k$  de 1**)

Affecter à  $U$  la valeur  $5 \cdot (3/5)^k$

Fin de Tant\_que

### Sortie

Afficher « la plus petite valeur de  $N$  est  $N_0 =$  »

Afficher  $k$

A suivre