

TD - Algorithmes Suites

Exercice 1. Termes d'une suite définie par une relation de récurrence

CODE

```
function suite($N) {
    $u=500;
    $n=0;
    while ($n<=$N) {
        $n=$n+1
        $u= 0.9*$u+1
        return [$n, $u]
    }
}
```

Pseudo Code

Algo suite(N)

$u = 500$

$n = 0$

Tant que $n \leq N$ Faire

$n \leftarrow n + 1$

$u \leftarrow 0.9 \times u + 1$

Fin Tant que

Retourner n, u

1. Compléter le tableau suivant afin d'évaluer la valeur des variables n et u pour chaque itération si on choisit $N = 5$.

u	500						
n	0						
Condition $n \leq N = 5$	VRAIE						

Donc la fonction va envoyer :, si on écrit **suite(5)** dans la console. Vérifier-le.

2. Expliciter la suite associée à cet algorithme. Pour n entier naturel : $u_0 = \dots$
 $u_{n+1} = \dots$
3. On remarque que le programme 1 ne renvoie pas le terme de rang 5 mais le suivant, modifier-le pour corriger ce problème.
4. Modifier le programme afin d'afficher les termes de la suite de rang 1 à N (avec les rangs associés). Vous devrez utiliser une boucle *while*.
5. Écrire un autre programme qui affiche les termes de la suite de rang 0 à N (avec les rangs associés) en utilisant cette fois une boucle "for i in range(...)":

CODE

```
function suite(N) {
    $u=500
    for ($i ...) {
        $u=...
        echo $i . ' ' . $u . '<br>';
    }
}
```

Exercice2. Suites définies par une relation de récurrence

Un couple fait un placement au taux annuel de 2 % dont les intérêts sont capitalisés tous les ans. Le couple a placé le montant de 1 000 euros à l'ouverture le 1^{er} janvier 2010 puis, tous les ans à chaque 1^{er} janvier, verse 2 400 euros. On note u_n le capital présent sur le compte le 1^{er} janvier 2010 + n après le versement annuel. On a donc $u_0 = 1000$ et on admet que pour tout entier n , la suite (u_n) est définie par :

$$\begin{aligned} u_0 &= 1000 \\ u_{n+1} &= 1.02 \times u_n + 2400 \end{aligned}$$

1. Calculer les 6 premiers termes de la suite.

$$u_0 = \dots ; u_1 = \dots ; u_2 = \dots ; u_3 = \dots ; u_4 = \dots ; u_5 = \dots$$

2. Calcul du terme d'indice donné.

L'algorithme suivant cherche à déterminer la somme présente sur le compte après un certain nombre d'années. Compléter et tester le programme avec quelques valeurs de n .

```
# CODE
function u($N) {
    $u = 1000; $n = 0;
    while $n < $N:
        $n = $n + 1
        $u = ...;
    return $u
}
```

Pseudo Code
 Algorithme u(N)
 $u, n \leftarrow 1000, 0$
 Tant que $n \leq N$ Faire
 $n \leftarrow n + 1$
 $u \leftarrow \dots$
 Fin Tant que
 retourner u

3. Problème de seuil.

L'objectif du couple est de constituer un capital de 18 000 euros.

On cherche donc maintenant à écrire un algorithme qui renvoie l'indice et la valeur du premier terme de la suite qui dépasse un 18 000. Compléter et tester le programme en déterminant l'année durant laquelle le couple aura atteint son objectif. Contrôler le résultat avec la fonction de la question précédente.

Code

```
function recherche($seuil) {
    $u = 1000;
    $n = 0;
    while ( ... ) { # à compléter
        $n = $n + 1;
        $u = ... : # à compléter
        return [$a, $u];
    }
}
```

Pseudo Code
 Fonction recherche(seuil)
 $u \leftarrow 1000$
 $n \leftarrow 0$
 Tant que Faire
 $n \leftarrow n + 1$
 $u \leftarrow \dots$
 Fin Tant que
 retourner (n, u)

u	1 000								
n	0	1	2	3	4	5	6	7	8
Condition ...	VRAIE								
Année	2010	2011							