Suites et algorithmes

Exercice 1

Soit la suite (u_n) définie par $u_0 = 150$ et pour tout entier naturel n :

$$u_{n+1} = 0{,}8 \cdot u_n + 45$$

1. Calculer u_1 et u_2.

2. Voici deux propositions d'algorithmes :

```
U ← 150
N ← 0
Tant que U⩾220
    U ← 0,8×U+45
    N ← N+1
Fin Tant que
```
Algorithme 1

```
U ← 150
N ← 0
Tant que U<220
    U ← 0,8×U+45
    N ← N+1
Fin Tant que
```
Algorithme 2

On s'intéresse, à la fin de son exécution, à la valeur de la variable `N` de l'algorithme.

a. Un seul de ces algorithmes permet d'affecter à la variable `N`, en fin d'exécution, le plus petit entier naturel n tel que $u_n \geqslant 220$.
Préciser lequel en justifiant pourquoi l'autre algorithme ne le permet pas.

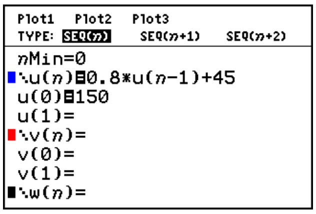
b. Quelle est la valeur de la variable `N` à la fin de l'exécution de cet l'algorithme?

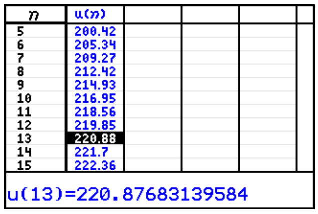
Correction 1

1. Voici les deux termes demandés :
   - $u_1 = 0{,}8 \times u_0 + 45 = 0{,}8 \times 150 + 45 = 120 + 45 = 165$
   - $u_2 = 0{,}8 \times u_1 + 45 = 0{,}8 \times 165 + 45 = 132 + 45 = 177$

2. a. C'est l'algorithme **2** qui permet d'affecter, en fin d'exécution, à la variable `N` le plus petit entier naturel n vérifiant $u_n \geqslant 220$
L'algorithme **1** affecte, en fin d'exécution, la valeur `0` à la variable `N`. En effet, la condition `U⩾220` n'est pas vérifiée pour la valeur `150` initiale de la variable `U` : la boucle ne sera jamais exécuté et le rang vaudra 0.

b. Par génération des termes de la suite (u_n), à l'aide de la calculatrice, on observe que le premier terme de la suite ayan une valeur supérieur ou égale à 220 a pour rang 13





Exercice 2

On considère l'algorithme suivant :

```
a ← 2
Pour i allant de 0 à 5
    a ← a×2
Fin Pour
```

1. Lors de son exécution pas à pas, indiquer les différentes valeurs prises par la variable `a`

2. Parmi les exrpressions choisies qu'elle*(s)* peuvent être l'expression d'une suite (u_n) afin que ses six premiers termes soient les valeurs prises par la variable `a` lors de l'exécution de l'algorithme précédent :

a. $u_n = 2 \cdot n, \quad \forall n \in \mathbb{N}$

b. $u_n = 2^n, \quad \forall n \in \mathbb{N}$

c. $u_n = 2^{n+1}, \quad \forall n \in \mathbb{N}$

d. $\begin{cases} u_0 = 2 \\ u_{n+1} = 2 \cdot u_n, \ \forall n \in \mathbb{N} \end{cases}$

e. $\begin{cases} u_0 = 2 \\ u_n = 2 \cdot u_{n+1}, \ \forall n \in \mathbb{N} \end{cases}$

f. $\begin{cases} u_0 = 2 \\ u_n = 2 \cdot u_{n-1}, \ \forall n \in \mathbb{N}^* \end{cases}$

Correction 2

1. Voici synthétisé ci-dessous, le fonctionnement de l'algorithme :

i	a
0	4
1	8
2	16
3	32
4	64
5	128

Ainsi, les différentes valeurs affectées à la variable `a` sont :
2 ; 4 ; 8 ; 16 ; 32 ; 128

2. Les suites définissant la suite (u_n) sont :
   - $\begin{cases} u_0 = 2 \\ u_{n+1} = 2 \cdot u_n, \quad \forall n \in \mathbb{N} \end{cases}$
   - $\begin{cases} u_0 = 2 \\ u_n = 2 \cdot u_{n-1}, \quad \forall n \in \mathbb{N} \end{cases}$
   - $u_n = 2^{n+1}, \quad \forall n \in \mathbb{N}$

Exercice 3

On considère l'algorithme suivant :

```
a ← −1
Pour i allant de 0 à 4
    a ← a×2−i+1
Fin Pour
```

1. Donner les différentes valeurs prises par la variable `a` lors d'une exécution pas à pas de cet algorithme.

2. Donner l'expression d'une suite (u_n) dont les cinq premiers termes sont les différentes valeurs prises par la variable `a` lors de l'exécution de cet algorithme.

Correction 3

1. Voici synthétisé ci-dessous, le fonctionnement de l'algorithme :

`i`	`a`
	-1
0	-1
1	-2
2	-5
3	-12
4	-27

Ainsi, les différentes valeurs affectées à la variable `a` au cours de l'exécution de l'algorithme sont :
$-1 \quad ; \quad -1 \quad ; \quad -2 \quad ; \quad -5 \quad ; \quad -12$

2. En regardant le fonctionnement de l'algorithme, en définissant la suite (u_n), on a :

$$u_0 = -1 \quad ; \quad u_{n+1} = 2u_n - n + 1 \quad \text{pour tout } n \in \mathbb{N}$$

