

# LES SUITES NUMÉRIQUES E06C

## EXERCICE N°1 Lecture graphique

On a représenté ci-contre une fonction  $f$ .

On définit une suite  $u$  par :  $\forall n \in \mathbb{N}, u_n = f(n)$

On admet que  $u_0 = 1$ .

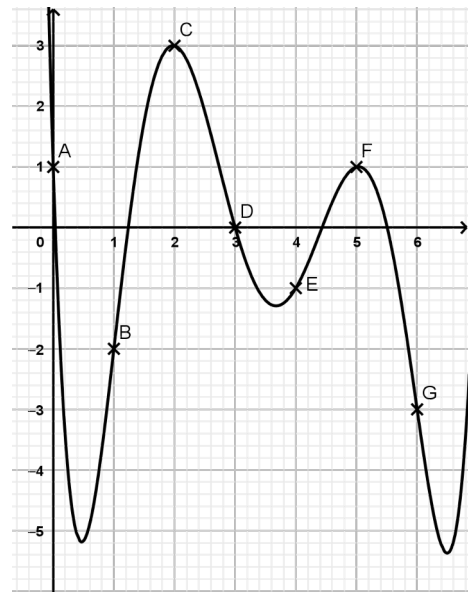
Donner les valeurs des six termes suivants.

Les termes de la suite sont représentés par les points.

$A(0, 1)$  correspond à  $u_0 = 1$  et ainsi de suite.

$$u_1 = -2 ; u_2 = 3 ; u_3 = 0$$

$$u_4 = -1 ; u_5 = 1 \text{ et } u_6 = -3$$



# LES SUITES NUMÉRIQUES E06C

## EXERCICE N°2 Utiliser un graphique (méthode à connaître)

On a représenté une fonction  $g$  ainsi que la droite d'équation  $y = x$  dans le graphique ci-contre.

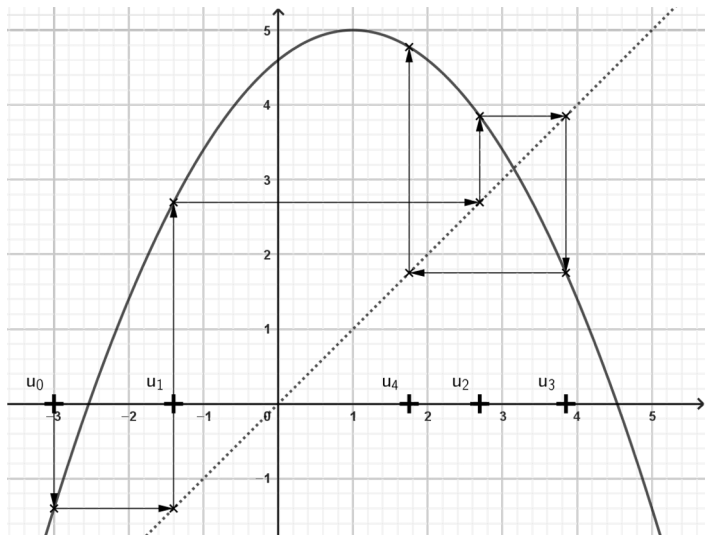
On définit la suite  $v$  par :

$$\begin{cases} v_0 = -3 \\ \forall n \in \mathbb{N}, v_{n+1} = g(v_n) \end{cases}$$

Déterminer les cinq premiers termes de la suite  $v$ .

$$u_0 = -3 ; u_1 \approx -1,4 ; u_2 \approx 2,7$$

$$u_3 \approx 3,8 \text{ et } u_4 \approx 1,7$$



# LES SUITES NUMÉRIQUES E06C

## EXERCICE N°3 Un peu de python

Soit  $(u_n)_{n \in \mathbb{N}}$  définie par :  $u_0 = 5$  et  $u_{n+1} = -u_n + 4$ .

On considère l'algorithme ci-contre :

1) Que permet d'afficher cet algorithme ?

Il permet d'afficher la valeur de  $u_{25}$ .

2) Quelle valeur affiche cet algorithme ?

$$u_{25} = -1$$

3) Modifier cet algorithme pour qu'il affiche la valeur de  $u_{40}$

On écrit 40 à la place de 25.

4) Coder cet algorithme en Python.

```
1 u = 5
2 for i in range(1,26): #Ne pas oublier la petite subtilité
3     u = -u+4
4 print(u)
```

```
u ← 5
Pour i allant de 1 à 25
    u ← -u+4
Fin pour
Afficher u.
```

# LES SUITES NUMÉRIQUES E06C

## EXERCICE N°4 *Encore un peu de python*

Soit  $u$  la suite définie par 
$$\begin{cases} u_0 = 5 \\ \forall n \in \mathbb{N}, u_{n+1} = 2u_n + 1 \end{cases}$$

1) Calculer  $u_1$  et  $u_2$ .

- $u_1 = 2u_0 + 1 = 2 \times 5 + 1$ , ainsi  $u_1 = 11$
- $u_2 = 2u_1 + 1 = 2 \times 11 + 1$ , ainsi  $u_2 = 23$

2) Écrire un algorithme permettant de calculer  $u_{20}$ .

```
u ← 5
Pour i allant de 1 à 20
    u ← 2 × u + 1
Fin pour
Afficher u
```

3) Coder cet algorithme en Python et l'utiliser pour calculer  $u_{20}$

```
1 u = 5
2 for i in range(1,21): #Ne pas oublier la petite subtilité
3     u = 2*u+1
4 print(u)
```