

On définit la suite de nombre suivante : $12 \Rightarrow 14 \Rightarrow 18 \Rightarrow 26 \Rightarrow 42 \Rightarrow 74 \Rightarrow \dots$

Le premier de ces nombres est donc $u_1 = 12$. Le nombre suivant s'obtient en multipliant le précédent par 2 et en soustrayant 10. Ainsi pour le 2nd terme de cette suite, on a $12 \times 2 - 10 = 14$ et donc $u_2 = 14$. Pour le 3^{ème}, on a : $14 \times 2 - 10 = 18$.

1 – Calculer le nombre qui suit le 6^{ème} terme égal à $u_6 = 74$:

$$u_7 = 74 \times 2 - 10 = 148 - 10 = 138$$

2- Le code ci-dessous est incomplet. La fonction *iter()* permet de calculer et retourner le nième terme en utilisant une méthode itérative. La fonction *rec()* fait la même chose avec une méthode récursive. Compléter ce code :

```

1 def iter(n) :
2     u = 12
3     for i in range(1,n) :
4         u = 2*u - 10
5     return u
6
7
8 def rec(n) :
9     if n == 1 : return 12
10    else :
11        return rec(n-1) * 2 - 10
12
13
14 assert iter(4) == 26
15 assert rec(4) == 26

```

3- Donner le contenu des différentes variables lorsque ce code est exécuté : sous forme de tableau pour les variables de la fonction *iter()*, sous forme de diagramme pour la fonction *rec()* :

Exécution de <code>iter(4)</code>		
n	u	i
4	12	1
	$2 \times 12 - 10 = 14$	2
	$2 \times 14 - 10 = 18$	3
	$2 \times 18 - 10 = 26$	

Valeur retournée : 26

Diagramme de la fonction *rec()* :

- Appel *rec(4)* : Valeur retournée : 26
- Appel *rec(3)* : $rec(3) \times 2 - 10 = 18 \times 2 - 10 = 26$
- Appel *rec(2)* : $rec(2) \times 2 - 10 = 14 \times 2 - 10 = 18$
- Appel *rec(1)* : $rec(1) \times 2 - 10 = 12 \times 2 - 10 = 14$