

Compétence **calculer** : suites et algorithmes

Exercice :

Soit la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_{n+1} = u_n + 2n + 1$ et $u_0 = 0$.

On note $S_n = \sum_{k=0}^n u_k$ soit $S_n = u_0 + u_1 + \dots + u_n$.

1. Mise en place sur le **tableur**

(a) À l'aide de deux formules adaptées au **tableur**, compléter le tableau suivant :

	A	B	C
1	n	u(n)	S(n)
2	0	0	0
3	1		
4	2		
5	3		
6	4		
7	5		

- formule en cellule B3 :
- formule en cellule C3 :

(b) Que pouvez-vous conjecturer ? Vous noterez \mathcal{P}_n votre proposition.

2. Mise en place d'**algorithmes** : u représente un terme de la suite (u_n) et S un terme de la suite de la somme (S_n) .

(a) boucle déterminée :

On souhaite calculer les valeurs de u et de S au rang $n = 5$ de la suite, le rang n est déterminé, connu.

Compléter l'algorithme suivant :

$u \leftarrow 0$

$S \leftarrow 0$

Pour n allant de 1 à 5

$u \leftarrow \dots$

$S \leftarrow \dots$

Fin Pour Vérifier votre algorithme en l'exécutant sur **Python** :

```
#algorithme boucle déterminée

u=0
S=0
for n in range(1,6):
    u=...
    S=...
#pour afficher on utilise la commande print
print('u(5)=' , u, 'S(5)=' , S)
```

(b) boucle indéterminée :

On souhaite déterminer le premier terme n pour lequel u sera supérieur à 1000 valeurs de u , le rang n est indéterminé, inconnu.

Compléter l'algorithme suivant :

```
n ← 0
u ← 0
S ← 0
Tant que .....
    n ← 0
    u ← .....
    S ← .....
```

Fin Tant que Vérifier votre algorithme en l'exécutant sur **Python** :

```
#algorithme boucle déterminée
n=0
u=0
S=0
while ... :
    n=...
    u=...
    S=...
#pour afficher on utilise la commande print
print('u(',n,')=',u,'S(',n,')=',S)
```

Exercice à faire à la maison :

1. Démontrer la proposition \mathcal{P} par récurrence.
2. Démontrer la proposition \mathcal{P} d'une autre manière, en utilisant la somme membre à membre de $u_{k+1} - u_k = 2n + 1$ pour k variant de 0 à $n - 1$.