

Boucle bornée - Corrigé

NUMWORKS

1 Exercice

Écrire une fonction `somme(n)` qui prend un nombre entier en argument et renvoie la somme des n premiers entiers ($1 + 2 + 3 + \dots + n$) et qui contient une boucle `for`.

1.1 Analyse de l'énoncé

On cherche à utiliser une boucle `for` dans la fonction `somme(n)`. Il faut donc tout d'abord repérer l'opération répétitive et déterminer combien de fois elle est répétée.

A la main, on peut donc commencer en testant les cas $n=1$, $n=2$, $n=3$, ...

– $n = 1$:

$$S(1) = 1$$

– $n = 2$:

$$S(2) = 1 + 2 = 3$$

– $n = 3$:

$$S(3) = 1 + 2 + 3 = 6$$

On voit bien en continuant $n=4$, $n=5$, ... qu'il est préférable d'ajouter 4, 5, ... au résultat précédent plutôt que tout recalculer : on préférera donc faire $S(4) = S(3) + 4 = 6 + 4 = 10$ que $S(4) = 1 + 2 + 3 + 4$.

On a distingué ici l'étape répétitive : " $S(i) = S(i-1) + i$ ".

Pour calculer la somme des entiers de 1 à n , il faudra faire varier i de 1 à n . On utilisera donc `range(1, n+1)`.

1.2 Résolution

On note S la somme cherchée. Au début de la boucle, la variable S vaut zéro : $S = 0$. A la fin de la boucle, on veut que $S = 1 + 2 + \dots + n$.

```
deg PYTHON
def puissance(x,n):
    resultat=1
    for i in range(n):
        resultat=resultat*x
    return resultat
```

FIGURE 1 – image 2

```
deg PYTHON
def somme(n):
    S=0
    for i in range(1,n+1):
        S=S+i
    return S
```

Avant la boucle `for`, on crée donc la variable `S` en l’initialisant à 0. A chaque itération de la boucle `for`, on réalise l’opération répétitive suivante : “stocke `S+i` dans `S`”.

1.3 Remarque pour finir

Il existe une formule explicite en fonction de `n` pour calculer la somme des `n` premiers entiers :

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

2 Un autre exercice à traiter

Écrire une fonction `puissance(x,n)` qui prend un réel `x` et un entier naturel `n` en argument et renvoie la puissance `n`-ième de `x`, x^n .

2.1 Corrigé

2.2 Quelques remarques sur le corrigé

Comme on répète n fois $x \times x \times \dots \times x$, on utilise une boucle **for** tout comme la somme des n premiers entiers. Ici on multiplie le résultat par x à chaque itération.

```
deg PYTHON
>>> 2**3
8
>>> |
```

En Python, il y a la possibilité d'utiliser une double étoile pour calculer une puissance.

```
deg PYTHON
>>> from math import *
>>> pow(2,3)
8.0
>>> |
```

Aussi, le module **math** de Python dispose d'une fonction **pow(x,n)** pour calculer la puissance d'un nombre.