TP # 4

Boucle `for` : Sommes et produits

Calculs de sommes et produits simples

- Pour calculer une somme $\sum_{k=m}^{n} a_k$:

```
S = 0 # avant la boucle, S = 0 ("somme vide")

for k in range(m,n+1) : # boucle pour k variant entre m et n
    S = S + a_k # on ajoute le k-eme terme

print(S) / return(S) # on affiche / on renvoie la valeur finale
```

- Pour calculer un produit $\prod_{k=m}^{n} a_k$:

```
P = 1 # avant la boucle, P = 1 ("produit vide")

for k in range(m,n+1) : # boucle pour k variant entre m et n
    P = P * a_k # on multiplie par le k-eme terme

print(P) / return(P) # on affiche / on renvoie la valeur finale
```

Exercice 1

1. Compléter le programme suivant pour afficher la valeur de la somme $\sum_{k=0}^{100}(2k+1)$.

```
S = .....

for k in .................... :

        S = ...................

print(S)
```

2. Calculer cette somme (avec des maths !) et vérifier le résultat annoncé par Python.

Exercice 2

1. Compléter la définition de la fonction `factorielle`

qui prend en entrée un entier $n \in \mathbb{N}$ et renvoie la valeur de $n! = \prod_{k=1}^{n} k$.

(Attention : la fonction doit marcher également pour $n = 0$)

```
def factorielle(n) :

        P = ....

        for k in ...................... :

            P = ................

        return P
```

2. Tester cette fonction pour calculer $10! =$ ....

Exercice 3

Pour tout $n \in \mathbb{N}^*$, soit $S_n = \sum_{i=1}^{n} \frac{1}{i^2}$. On veut montrer : $\lim_{n\to+\infty} S_n = \frac{\pi^2}{6}$.

1. Définir une fonction `somme` qui prend en entrée $n \in \mathbb{N}^*$ et renvoie la valeur S_n.

2. A l'aide de Python, compléter avec des valeurs approchées :

$$\frac{\pi^2}{6} \simeq \text{...}$$

$S_1 =$ ......, $S_{10} \simeq$ .... , $S_{100} \simeq$ .... , $S_{10000} \simeq$ ....

Remarque 1

Pour renvoyer la valeur finale :

- A l'intérieur d'une fonction on utilisera systématiquement $\boxed{\texttt{return}}$.
- En dehors d'une fonction (comme dans l'exercice 1), on utilisera plutôt $\boxed{\texttt{print}}$.

Sommes doubles

- Pour calculer une somme double $\sum_{i=m}^{n}\left(\sum_{j=q}^{p} a_{i,j}\right)$, on utilise deux boucle "imbriquées" :

```
S = 0 # avant la boucle, S = 0 ("somme vide")

for i in range(m,n+1) : # boucle pour i variant entre m et n
    for j in range(q,p+1) : # boucle pour j variant entre q et p
        S = S + a_(i,j) # on ajoute le terme correspondant

print(S) / return(S) # on affiche / on renvoie la valeur finale
```

Ceci fonctionne aussi bien pour des sommes "libres" $\sum_{i=1}^{n}\sum_{j=1}^{p} a_{i,j}$

que pour des sommes "avec contraintes" de type $\sum_{i=1}^{n}\sum_{j=i}^{n} a_{i,j}$ ou $\sum_{j=1}^{n}\sum_{i=1}^{j} a_{i,j}$ ...

Exercice 4

1. Ecrire la somme double suivante comme une "somme de somme" :

$$\sum_{\substack{1\leqslant i\leqslant 5\\ 1\leqslant j\leqslant 10}} \frac{1}{i+j} =$$

2. En déduire comment compléter le programme suivant pour qu'il affiche la valeur de cette somme double.

```
S = ........

for i in .................... :

        for j in ..................... :

                S = ............................

print(S)
```

Valeur approchée : $\sum_{\substack{1\leqslant i\leqslant 5\\ 1\leqslant j\leqslant 10}} \frac{1}{i+j} \simeq$ ....

Exercice 5

1. Ecrire la somme double suivante comme une "somme de somme" :

$$\sum_{1\leqslant i\leqslant j\leqslant n} 2^{i+j} =$$

2. Définir une fonction qui prend en entrée n et renvoie la valeur de la somme double précédente.