DM2

Exercice 1 :

```
1     Algorithme A1
2     Variables
3     mntPaye , prixCafe, monnaie: réels
4
5     DEBUT :
6     Afficher "Prix Café sélectionné :"
7     lire prixCafe
8     Afficher "Montant payé : "
9     lire mntPaye
10
11
12    SI mntPaye > prixCafe ALORS
13       monnaie ← mntPaye - prixCafe
14       Afficher "Monnaie rendue ."
15       Afficher monnaie
16    FIN_SI
17    FIN
```

1) Que fait cet algorithme ?

2) Faire un test avec comme entrée au clavier

a . 1,5 puis 2 b. 1,5 puis 1

```
1     Algorithme A2
2     Variables
3     mntPaye , prixCafe, monnaie: réels
4     ilyaReste : ……
5     DEBUT :
6     Afficher "Prix Café sélectionné : "
7     lire prixCafe
8     Afficher "Montant payé : "
9     lire mntPaye
10
11    ilyaReste ← (mntPaye > prixCafe)
12    SI …… ALORS
13       monnaie ← mntPaye - prixCafe
14       Afficher "Monnaie rendue ."
15       Afficher monnaie
16    FIN_SI
17    FIN
```

3) On veut modifier l'algorithme A1 en A2

a. A l'aide de la ligne 11 de A2 , quel est le type de variables de `ilyaReste` ?

b. Compléter alors la ligne 12

4) Que se passe-t-il dans le test 2)b. . Modifier l'algorithme A1 en A3 **pseudocode** pour corriger ce problème.

5) Implémenter en Python l'algorithme A3

Exercice 2 :

Partie A : Pour calculer la somme des n nombres entiers : on peut utiliser la fonction :

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

1) **Somme des *n* premiers entiers Non Nul :**

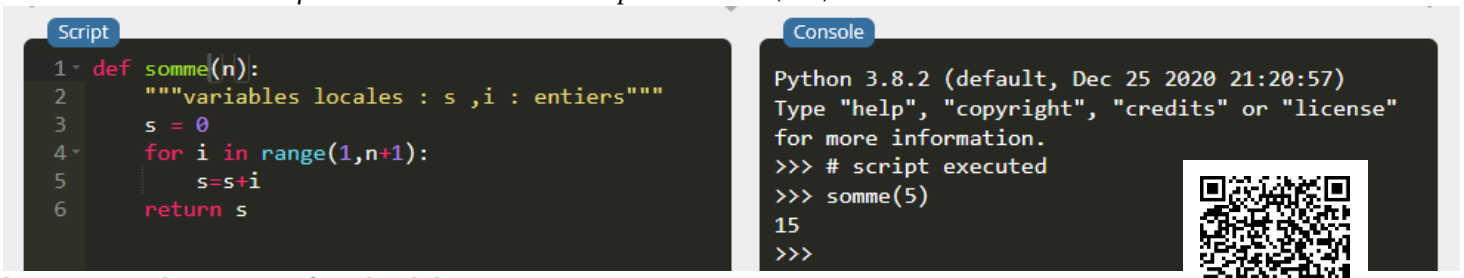
a. Ecrire cette fonction somme en pseudo Code.

b. Faire un test pour n =5

c. Que vaut `somme(10)` puis `somme(1000)`, `somme(2000)` ?

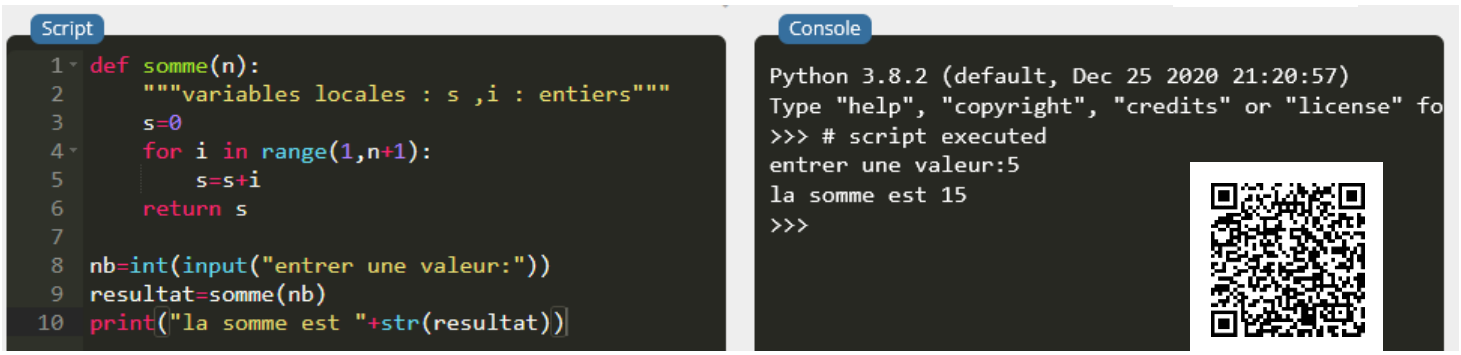
Aide : 1. *On écrit cette fonction la console Python ou dans un script que l'on exécute, pour charger la fonction.*

2. *On exécute la fonction dans console en tapant* `somme(10)`



https://capytale2.ac-paris.fr/web/c/bd70-814155

Ou



https://capytale2.ac-paris.fr/web/c/936c-814296

2) **Somme des *n*+1 premiers entiers pairs :**

a. Donner le code en Python d'une fonction `sommeP` qui prend en entrée un nombre entier *n* et retourne la somme des *n*+1 **premiers** entiers **pairs**.

$S = 2\times\mathbf{0} + 2\times1 + 2\times2 + \ldots + 2\times i + \ldots + 2\times\boldsymbol{n}$

Aide : ¤ *Si un nombre est pair il s'écrit* `2*i`
¤ *On doit choisir entre les boucles* `Pour` *ou* `TantQue`

b. Quelle est la valeur de `sommeP(500)` , `sommeP(1000)` ?

c. Y a-t-il un lien entre les fonctions `somme` et `sommeP` ? pour toutes les entrée *n* ?

d. Donner le code en Python d'une fonction `sommeP2` qui prend en entrée un nombre entier *n* et retourne la somme des **premiers** entiers **pairs** inférieur à *n* : (On pourra utiliser un `while`)

$S = 2\times\mathbf{0} + 2\times1 + 2\times2 + \ldots + 2\times i + \ldots + 2\times\boldsymbol{k}$ avec $2\times\boldsymbol{k} < \boldsymbol{n}$

Quelle est la valeur de `sommeP2(500)` ?

3) Somme des carrés : $S_n = 1 + 2^2 + 3^2 + \ldots + n^2$

a. Donner le code en Python d'une fonction `carre` qui prend en entrée un nombre entier *n* et retourne la somme des carrés des *n* nombres entiers.

b. Donner `carre(10)` , `carre(100)`

Partie B :

Définition : Soit un entier naturel *n*. On définit la factorielle de *n*, définie par : $n! = n\times(n-1)\times\ldots\times3\times2\times1$.

Application : La fonction factorielle est souvent utilisée en dénombrement, afin de calculer des probabilités.

Exemple : le nombre de placements possibles de 4 personnes sur un banc vaut : $4! = 4\times3\times2\times1 = 24$.

On demande d'écrire une fonction **`fact`**, qui prend en argument un entier, et qui retourne la valeur de la factorielle de cet entier.
L'algorithme donné ci-contre appelle cette fonction pour calculer la factorielle d'un nombre entier entre au clavier. Il faut donc :

(a) Calculer 5 ! =`fact(5)` et 6 ! =`fact(6)` en détaillants les calculs

(b) On remarque que $5! = 5\times(4!)$ et $6! = 6\times(5!)$ soit $(n+1)! = (n+1)\times(n!)$

Compléter le corps de la fonction, en utilisant la définition et s'aidant de la partie A (ici on a la multiplication au lieu de l'addition)

(c) Écrire un script en Python qui met en œuvre cet algorithme.

(d) Que vaut `fact(10)` et `fact(15)` ?

```
ALGORITHME  factorielle :

FONCTION fact(n : entier) : entier
p ← ...          # initialisation
POUR i allant de 1 à n faire :
     p ← p×...
renvoyer ...

FIN_FONCTION

DEBUT
Variable nb: entier
Afficher "Saisir l'entier N : "
Lire nb
Afficher "N! = ", fact(nb)
FIN
```