

## NOMS - PRENOMS DU GROUPE :

## Exercice 1. Un premier programme sous Python

## Code Python

```

VarA=2 # la variable VarA vaut 2
b4=3.1 # la variable b4 vaut 3,1
s=b4+3 # s vaut 6,1
c,d=4.2 , 7 # c vaut 4,2 et d vaut 7
e= " bonjour" # e vaut " bonjour" (chaîne de caractères)
E= "Hello!" # E vaut "Hello!" (chaîne de caractères différente de e car minuscules et majuscules sont considérées différentes)

```

Si on lance le programme précédent, rien ne s'affiche car aucun affichage n'est demandé.

Par contre, si vous écrivez VarA dans la console de droite (sur repl.it par exemple), l'affichage sera 2 ...

## Module et console

La copie d'écran suivante montre, à gauche, un module dont l'exécution est lancée par un clic sur le bouton < ; à droite on voit la console, où on exécute quelques instructions pour tester la correction du module. Les commandes de la console sont annoncées par l'invite >. En revanche, le module est un simple fichier texte, qui pourrait être retravaillé dans n'im- porte quel éditeur de texte.

The screenshot shows a Python REPL interface. On the left, the code from the exercise is pasted into a text editor. On the right, the console shows the output of the code execution. The output is as follows:

```

Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
> VarA
=> 2
> E
=> 'Hello !'
> c
=> 4.2
>

```

## Exercice 2. Variables

```

1 a=2
2 b=-5
3 a=a+b
4 b=a-b
5 print("Maintenant, a=",a," et b=",b)

```

On considère l'algorithme ci-dessus écrit sous Python.

Donner l'affichage de la ligne 5.

Saisir ensuite le programme sous Python pour vérifier vos résultats.

## Exercice 3. Suite d'affectations

On considère l'algorithme suivant écrit en pseudo code :

**Traitement :**

```
L1          U ← 500
L2          N ← 0
L3          U ← 0.7 × U + 300
L4          N ← N + 1
```

1. Compléter le tableau suivant afin de déterminer les valeurs affichées en sortie.

Ligne	U	N
L1	...	...
L2	...	...
L3	...	...
L4	...	...

2. Écrire sous Python et vérifier votre tableau.

## Exercice 4. Une fonction d'Euler



**def nom\_ fonction(paramètres) :**

*def nom\_ fonction(paramètres) :* définit une nouvelle fonction, les deux points entraînent une indentation délimitant la déclaration de la fonction. Le bloc peut servir à effectuer une série d'actions, mais le plus souvent il se termine par *return* pour renvoyer une ou plusieurs valeurs.



**Code Python**

```
def f(x):
    return x * x - x + 41
```

1. La fonction définie ci-dessus renvoie l'image de la variable  $x$  par la fonction  $f$  définie par  $f(x) = x^2 - x + 41$ .

Tester le programme avec des entiers naturels en écrivant dans la console directement  $f(1)$  ou  $f(2)$  par exemple.

Donner les valeurs de  $f(20)$  et  $f(35)$  à l'aide du programme.

2. On veut maintenant calculer les valeurs de la fonction  $f$  pour  $x$  entier variant de 0 à 19 par exemple.



**range(début , fin , pas)**

*range(début , fin , pas)* : Génère une liste d'entiers les paramètres *début* et *pas* sont optionnels.

- Dans l'intervalle  $[0 , fin[$  si un seul paramètre est renseigné.  
 $L = range(4)$  va créer la liste  $[0 , 1 , 2 , 3]$  de 4 termes, le premier sera  $L[0]=0$ , le dernier  $L[3] = 3$ .
- Dans l'intervalle  $[début ; fin[$  si 2 paramètres sont renseignés.  
 $L = range(1 , 5)$  va créer la liste  $[1 , 2 , 3 , 4]$  le premier terme sera  $L[0] = 1$  et le dernier  $L[3] = 4$ .
- Dans l'intervalle  $[début ; fin[$  mais de pas en pas, si les 3 paramètres sont renseignés.  
 $L = range(2 , 9 , 2)$  va créer la liste  $[2 , 4 , 6 , 8]$ .

```

Code Python
def f(n):
    return n**2 - n + 41 # n**2 retourne le carré de n c'est la notation puissance
valeursx = [ x for x in range(20) ] # permet d'itérer pour x de 0 à 19
valeursf = [ f(x) for x in range(20) ]
# on pouvait aussi écrire valeursf = [ f(x) for x in valeursx ]
    
```

Pour visualiser le résultat, écrire simplement valeursx et valeursf dans la console de droite.

Que remarque-t-on ?

.....

.....

.....

.....

3. Faire la même étude avec la fonction  $f$  définie par  $f(x) = 36x^2 - 2358x + 36809$

.....

.....

.....

.....