Initiation à l'informatique - 1er semestre

TP 1

Premiers programmes en Python

Exercice 1.1 *Les commandes élémentaires*
Reprendre les exercices du TD1.
Terminer en effaçant tous les répertoires utilisés précédemment pour créer une arborescence propre (suggestion : à partir de votre compte, avoir un répertoire `Initinfo`, puis 2 sous-répertoires `tp`, et `exoperso` puis se positionner dans `tp`).

Exercice 1.2 *Premiers pas en Python*
Le but de cet exercice est de tester et d'expliquer diverses expressions Python. Dans une fenêtre `xterm` lancez l'interpréteur Python à l'aide de la commande `python`. Pour chacune des expressions suivantes que vous taperez, expliquez le résultat fourni par l'interpréteur Python.

1. `5 + 7 * 10`
2. `13 / 4`
3. `13.0 / 4`
4. `13 % 4`
5. `i = 5`
6. `i`
7. `i = i + 4`
8. `i`
9. `j = 0`
10. `k`
11. `bonjour = salut`
12. `bonjour = "salut"`
13. `bonjour`
14. `i < j`
15. `p = (j <= 3)`
16. `p`
17. `p and (i == 9 )`
18. `p or (j <= 3)`
19. `i <> j`
20.
```
if j < 10:
    print j, " < 10"
else:
    print j, " >= 10"
```
21. `range(0,5)`
22. `range(5)`
23. `range(-3,4)`
24. `range(1,21,2)`
25.
```
for i in range(10):
    print i
```
26.
```
for i in range(2,11):
    print i,
```

Pour quitter l'interpréteur Python, tapez `C-d` (touche Ctrl en même temps que la touche D).

Exercice 1.3
En vous aidant de la feuille aide-mémoire, créez avec *Emacs* un fichier de nom `exo1-3.py` dans lequel vous taperez le programme ci-dessous. Celui-ci contient le code de la fonction `f` suivi de l'instruction permettant d'afficher le résultat de l'appel à cette fonction pour $x = 1$.

```
def f(x):
    resultat = 5 * x + 4
    return resultat
print f(1)
```

Exécutez le programme sous *Emacs* (voir feuille aide-mémoire). Ajoutez les instructions nécessaires pour faire afficher le résultat de cette fonction pour $x = 3$ et $x = 1000$
Dans le même fichier écrivez une fonction qui calcule $g(x) = 3x^2 + 2x + 7$. Affichez le résultat de cette fonction pour $x = 10$, $x = 20$, $x = 30$.
Calculer $3(5x + 4)^2 + 2(5x + 4) + 7$ pour $x = 2$

Exercice 1.4

Dans un fichier de nom `exo1-4.py` tapez et exécutez la fonction suivante :

```
def f(x, y) :
  if x < y :
    return x
  else :
    return y
```

Ajoutez les instructions nécessaires pour faire afficher le résultat de cette fonction pour $x = 3$ et $y = 4$. Dans le cas général que fait cette fonction ?
Qu'obtient-on si on exécute l'instruction :

```
print f(f(4,10),6)
```

Que pensez-vous de la fonction g suivante, à taper dans le même fichier

```
def g(x, y) :
  if x < y :
    return x
  return y
```

Exercice 1.5

Les fonctions des exercices 5 à 9 sont à sauvegarder dans le fichier `exo1-5a9.py`
Écrire une fonction `somme_carres(n)` qui renvoie

$$\sum_{i=1}^{n} i^2$$

Exercice 1.6

Ecrivez une fonction `suite` qui calcule le nième terme de la suite $U_0 = 2, U_n = 4U_{n-1} + 3$.
Affichez ensuite les 10 premiers termes de cette suite.

Exercice 1.7

Ecrire une fonction `median` qui retourne l'élément médian de 3 nombres passés en paramètres.

Exercices complémentaires

Exercice 1.8

Ecrire une fonction `une_minute_en_plus` qui affiche l'heure une minute après celle passée en paramètre sous forme de deux entiers que l'on suppose cohérents (voir td1)

Exercice 1.9

Ecrire une fonction `plus_petit_entier(a,n)` qui retourne le plus petit entier `k` tel que

$$a^k > n$$