

22 février 2008
Durée 1h20
Documents non autorisés

Exercice 1. Donner et justifier la valeur de l'expression : $x/y > 1$ and $x > 1$

pour $x = 3$ et $y = 2$,

pour $x = 10$ et $y = 6.0$,

pour $x = 10.0$ et $y = 8$,

pour $x = 0$ et $y = 2$

pour $x = 2$ et $y = 0$.

Corriger cette expression afin qu'elle ne produise jamais d'erreur.

Exercice 2. Donner et justifier la valeur des variables **a**, **b**, **c** et **k** après la suite d'instructions :

a = 3

b = 2.0

c = 3.0

k = 5

a = **a** * **k**/2

b *= **k**/2

c = **c** * **k**/2

Exercice 3. Un nombre entier **n** est divisible par 9 si la somme de ses chiffres est un multiple de 9.

Utiliser cette propriété pour écrire une fonction booléenne `multiple_de_neuf(n)` qui retourne `True` si **n** est un multiple de 9 et `False` sinon.

Vous ne devez pas utiliser d'opérateur ou de fonction qui calcule le quotient ou le reste d'une division par 9.

Exercice 4. On considère la suite définie par :

$$\begin{cases} u_0 = 2 \\ u_n = 3 * u_{n-1} - 1 \end{cases}$$

Ecrire :

1. une fonction itérative utilisant la boucle `while` qui calcule le terme de rang **n** de la suite,
2. une fonction itérative utilisant la boucle `for` qui calcule le terme de rang **n** de la suite,
3. une fonction récursive qui calcule le terme de rang **n** de la suite,

4. une fonction qui, étant donné un entier m , calcule le rang du premier terme de la suite supérieur ou égal à m ,
5. une fonction qui renvoie un tableau contenant les n premiers termes de la suite.

Exercice 5. Écrire une fonction `nombre_occurrences(t, x)` qui, étant donné une *Liste-Python* t et un élément x , calcule le nombre d'occurrences de x dans t . Par exemple, le nombre d'occurrences de 2 dans `[4, 2, 2, 3, 4, 2]` est 3.

Exercice 6. Soit la fonction f suivante prenant comme paramètres deux *Liste-Python* :

```
def f(vect1, vect2):
    resultat = 0
    for i in range (len(vect1)):
        resultat += vect1[i]*vect2[i]
    return resultat
```

Quelles sont les valeurs successives de `resultat` et `i` lorsqu'on appelle f avec :

`t1 = [2, 4, 1, 7, 3]` et `t2 = [6, 2, 5, 2, 3]`,
`t1 = [2, 4, 1]` et `t2 = [6, 2, 5, 2, 3]`,
`t1 = [6, 2, 5, 2, 3]` et `t2 = [2, 4, 1]`

Dans les trois cas, donner le résultat de l'appel de la fonction f en signalant les éventuels cas d'erreur.