Mathématiques supérieures le 29 Septembre 2025

Correction du test de positionnement

Durée : 2H
Calculatrice interdite

Ce sujet est composé de 7 exercices.

Exercice 1. *Instructions de base*

Donner le résultat des instructions suivantes :

1. `>>> 1+4`
2. `>>> 4*3`
3. `>>> 5/2`
4. `>>> 2**4`
5. `>>> 13%3`
6. `>>> 13//3`
7. `>>> 3+4*5`
8. `>>> (6-2)**(1/2)`

Correction.

1. `5`
2. `12`
3. `2.5`
4. `16`
5. `1`
6. `4`
7. `23`
8. `2`

Exercice 2. *Opérateurs de comparaison et opérateurs logiques*

Donner le résultat des instructions suivantes :

1. `>>> 1==4`
2. `>>> 2<=8`

3. `>>> (7*2)!=(16-2)`
4. `>>> 'abc'<'coucou'`
5. `>>> 'abc'<'abb'`
6. `>>> True and False`
7. `>>> False or True`
8. `>>> not 6<=3`
9. `>>> (1==6) or (2<4)`

Correction.

1. `False`
2. `True`
3. `False`
4. `True`
5. `False`
6. `False`
7. `True`
8. `True`
9. `True`

Exercice 3. *Affectation de variables*

Dans chacun des cas suivants, donner la valeur finale de la variable `x` :

1.
```
>>> x = 1
>>> y = 3
>>> x = y+2
```
2.
```
>>> x = 1
>>> x = x+2
```
3.

```
>>> x = 1
>>> x *= 2
```

4.

```
>>> x = 1
>>> y = x
>>> y = y-1
>>> x += 1
```

5.

```
>>> x = 'Coucou'
>>> x = x+' tout le monde !'
```

Correction.

1. `x` vaut `5`
2. `x` vaut `3`
3. `x` vaut `2`
4. `x` vaut `2`
5. `x` vaut `'Coucou tout le monde'`

Exercice 4. *Chaînes de caractères*

On considère la chaîne de caractères `"Super ce test, je m'amuse bien !"` contenue dans une variable `chaine` c'est-à-dire que de dans cette exercice, nous considérons comme effectuée l'affectation :

```
>>> chaine="Super ce test, je m'amuse bien !"
```

Déterminer le résultat des instructions suivantes :

1. `>>> chaine[0]`
2. `>>> chaine[4]`
3. `>>> chaine[-1]`
4. `>>> 3*chaine[1]`
5. `>>> len(chaine)`
6. `>>> chaine[1]+chaine[3]`
7. `>>> chaine[-11]+chaine[2]+chaine[-9]+chaine[-5]`

8. `>>> chaine[6:8]`
9. `>>> chaine[:5]`

Correction.

1. `'S'`
2. `'r'`
3. `'!'`
4. `'uuu'`
5. `32`
6. `'ue'`
7. `'mpsi'`
8. `'ce'`
9. `'Super'`

Exercice 5. *Boucles*

Dans chacun des cas suivants, donner la valeur finale de la variable `x` :

1.
```
x = 0
for i in range(4):
    x = x + i
```
2.
```
x = 1
while x <= 14:
    x *= 2
```

Correction.

1. `x` vaut `6`$= 0+0+1+2+3$ une fois la boucle finie.
2.
3. `x` vaut `16` une fois la boucle finie :
   — à la fin du premier tour, `x` vaut `2`$= 2\times 1$ et on a `x<=14`, la boucle continue ;
   — à la fin du deuxième tour, `x` vaut `4`$= 2\times 2$ et on a `x<=14`, la boucle continue ;

— à la fin du troisième tour, `x` vaut `8`$= 2 \times 4$ et on a `x<=14`, la boucle continue ;

— à la fin du quatrième tour, `x` vaut `16`$= 2 \times 8$ et on a `x>14` donc la boucle se termine et la valeur finale de `x` est bien `16`.

Exercice 6. *Une fonction mystère*

On donne le programme suivant :

```
def mystere(n):
    p=1
    for i in range(1,n+1):
        p=p*i
    return p
```

1. Que renvoie `mystere(4)` ?
2. Plus généralement, que renvoie `mystere(n)` pour `n` un entier naturel ?

Correction.

1. `mystere(4)` renvoie 24
2. `mystere(n)` renvoie $n!$ (on le prouvera correctement bientôt !)

Exercice 7. *Une suite à coder*

On considère la suite récurrente

$$\begin{cases} u_0 = 1 \\ u_{n+1} = u_n^2 + 1 \quad \text{pour } n \in \mathbb{N} \end{cases}$$

Écrire une fonction `suite(n)` qui renvoie la valeur de u_n pour n un entier naturel.

Correction.

```
def suite(n):
    u = 1
    for i in range(n):
        u = u**2+1
    return u
```