

Structures de contrôle – corrigés des exercices

Exercice 1

1. test
2. x et y
3. 5 et 4
4. somme
5. 20

Exercice 2

5 9 True

Exercice 3

1. Lignes complétées :

```
1 def bob_en_weekend(heure):
2     # 'mode' est une chaîne de caractères :
3     # soit "NSI", soit "console"
4     if heure > 0 and heure < 10:
5         mode = "console"
6     elif heure < 14:
7         mode = "NSI"
8     elif heure > 14 and heure < 24:
9         mode = "console"
10
11     print(mode)
```

2. Avec deux lignes de tests :

```
1 def bob_en_weekend(heure):
2     if (heure > 0 and heure < 10) or (heure > 14 and heure < 24):
3         mode = "console"
4     else:
5         mode = "NSI"
6
7     print(mode)
```

On aurait même pu écrire :

```
1 def bob_en_weekend(heure):
2     mode = "NSI"
3     if (heure > 0 and heure < 10) or (heure > 14 and heure < 24):
4         mode = "console"
5     print(mode)
```

3. Avec le jour de la semaine en plus :

```
6 def mode_bob(jour, heure):
7     # on admet que 'jour' est une chaîne de caractères :
8     # "lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi" ou "dimanche"
9     if jour == "samedi" or jour == "dimanche" or jour == "mercredi":
10         mode = "console"
11     else:
12         mode = "NSI"
13     print(mode)
```

Exercice 4

1. (a)

```
p = 1

1 < 100 ? True => tour de boucle 1
p = p * 2 = 1 * 2 = 2

2 < 100 ? True => tour de boucle 2
p = 4

4 < 100 ? True => tour de boucle 3
p = 8

8 < 100 ? True => tour de boucle 4
p = 16

16 < 100 ? True => tour de boucle 5
p = 32

32 < 100 ? True => tour de boucle 6
p = 64

64 < 100 ? True => tour de boucle 7
p = 128

128 < 100 ? False => On ne rentre pas dans la boucle
et on passe à l'instruction suivante (print).
```

(b) 7

(c) 64.0

Exercice 5

1. 2

2. 129

3. Rien, le programme ne finit jamais : on a créé une boucle infinie.

Exercice 6

La fonction `bar` se termine quelle que soit la valeur de `m`.

Exercice 7

Affichages produits :

```
0
2
4
6
8
.
.
.
2 * (n-1)
```

Nombre de tours de boucle effectués : `n`

Exercice 8

```
1. 16
2. def truc(n):
3     x = 1
4     for k in range(1, n):
5         x = 2 * x
6     print(x)
```

Exercice 9

```
1 def somme(n):
2     s = 0
3     for k in range(1, n+1):
4         s = s + k
5     print(s)
```

Exercice 10

10^{12} devrait être affiché après 10^{12} tours de boucle mais cette opération demande trop de ressources et, bien que la boucle soit bornée, le programme ne s'arrête pas (du moins sur nos ordinateurs ordinaires).