

Test n°1 – correction

Nom et prénom :

EXERCICE 1 : (2pt) Compléter les tableaux ci-contre.

a	b	$a \text{ et } b$	non a	$(a \text{ et } b) \text{ ou non } a$
0	0	0	1	1
0	1	0	1	1
1	0	0	0	0
1	1	1	0	1

a	b	non b	$(\text{non } b) \text{ et } a$	$(\text{non } b) \text{ ou } ((\text{non } b) \text{ et } a)$
0	0	1	0	1
0	1	0	0	0
1	0	1	1	1
1	1	0	0	0

EXERCICE 2 : (1,5pt)

- 1) Que vaut l'expression " $(a \text{ et non } b) \text{ ou } (b \text{ et } c)$ " lorsque $a = 0$, $b = 0$ et $c = 1$? Elle vaut 0.
- 2) Compléter les égalités suivantes pour que l'expression " $(\text{non } a) \text{ et } (\text{non } b) \text{ et } c$ " soit vraie :

$$a = 0, b = 0 \text{ et } c = 1$$

- 3) Compléter les égalités suivantes pour que l'expression " $a \text{ ou } (\text{non } b) \text{ ou } c$ " soit fausse :

$$a = 0, b = 1 \text{ et } c = 0$$

EXERCICE 3 : (3pt) Compléter les tableaux de valeur ci-dessous :

	a	b	c
$a = 2$	2		
$b = 8$	2	8	
$c = b - a$	2	8	6

	a	b
$a = 5$	5	
$b = a + 4$	5	9
$a = a + 1$	6	9

	a	b
$a = 4$	4	
$b = a * a$	4	16
$a = b - 4$	12	16

EXERCICE 4 : (2pt) On considère les fonctions suivantes :

```
def f1(x):
    y = 4*x - 5
    return y
```

```
def f2(x):
    a = 2*x
    b = 3*a + 1
    return b
```

```
def f3(x):
    m = 5*x
    n = x-6
    return m+n
```

```
def f4(x):
    return x-7
```

Compléter les résultats des appels suivants :

>>> f1(6) 19	>>> f3(2) 6
>>> k = 1 >>> f2(k+5) 37	>>> a = 9 >>> f1(f4(a)) 3

EXERCICE 5 : (1pt) Parmi les expressions suivantes, entourer celles qui sont des noms de variables valides en Python :

- 1) var_1_ABLE 2) trois*deux 3) bla bla 4) _16a17

EXERCICE 6 : (1pt) Dans chacun des cas, expliquer pourquoi il y a un message d'erreur :

```
>>> prix_initial = 20000
>>> prix_final = prxi_initial * 0.70
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
NameError: name 'prxi_initial' is not defined
```

Le nom `prxi_initial` n'est pas défini. C'est en fait `prix_initial` qu'il fallait écrire.

```
>>> def test (x, y):
    a = x + y
    b = x * y
    return a + b

File "<pyshell>", line 3
    b = x * y
IndentationError: unexpected indent
```

La première ligne de la fonction n'a pas la même indentation que les suivantes.

EXERCICE 7 : (1,5pt) On considère les deux fonctions ci-dessous :

```
def simple_au_double1(x):
    print(x)
    print(2*x)
```

```
def simple_au_double2(x):
    return x
    return 2*x
```

- 1) Indiquer les 2 expressions rentrées pour obtenir les résultats ci-dessous. Vous devez choisir parmi : `simple_au_double1(5)`, `simple_au_double2(5)`, `simple_au_double1(10)` ou `simple_au_double2(10)`

```
>>> simple_au_double1(10)
10
20
>>> simple_au_double2(10)
10
```

- 2) Entourer l'expression ci-dessous qui provoque une erreur :

- a) 1 + simple_au_double1(4) b) 1 + simple_au_double2(4)

Test n°1 – correction

Nom et prénom :

EXERCICE 1 : (2pt) Compléter les tableaux ci-contre.

a	b	$a \text{ et } b$	$\text{non } b$	$(a \text{ et } b) \text{ ou non } b$
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

a	b	$\text{non } a$	$(\text{non } a) \text{ et } b$	$(\text{non } a) \text{ ou } ((\text{non } a) \text{ et } b)$
0	0	1	0	1
0	1	1	1	1
1	0	0	0	0
1	1	0	0	0

EXERCICE 2 : (1,5pt)

- 1) Que vaut l'expression " $(a \text{ et non } b) \text{ ou } (b \text{ et } c)$ " lorsque $a = 0$, $b = 1$ et $c = 0$? Elle vaut 0.
- 2) Compléter les égalités suivantes pour que l'expression " $(\text{non } a) \text{ et } b \text{ et } (\text{non } c)$ " soit vraie :

$$a = 0, b = 1 \text{ et } c = 0$$

- 3) Compléter les égalités suivantes pour que l'expression " $a \text{ ou } b \text{ ou } (\text{non } c)$ " soit fausse :

$$a = 0, b = 0 \text{ et } c = 1$$

EXERCICE 3 : (3pt) Compléter les tableaux de valeur ci-dessous :

	a	b	c
$a = 3$	3		
$b = 5$	3	5	
$c = b - a$	3	5	2

	a	b
$a = 6$	6	
$b = a + 4$	6	10
$a = a + 1$	7	10

	a	b
$a = 5$	5	
$b = a * a$	5	25
$a = b - 4$	21	25

EXERCICE 4 : (2pt) On considère les fonctions suivantes :

```
def f1(x):
    y = 4*x - 5
    return y
```

```
def f2(x):
    a = 2*x
    b = 3*a + 1
    return b
```

```
def f3(x):
    m = 5*x
    n = x-6
    return m+n
```

```
def f4(x):
    return x-7
```

Compléter les résultats des appels suivants :

```
>>> f1(3)
7
>>> k = 1
>>> f2(k+3)
25
```

```
>>> f3(5)
24
>>> a = 11
>>> f1(f4(a))
11
```

EXERCICE 5 : (1pt) Parmi les expressions suivantes, entourer celles qui sont des noms de variables valides en Python :

- 1) une variable 2) p1_de_mie 3) _15h30 4) un+cing

EXERCICE 6 : (1pt) Dans chacun des cas, expliquer pourquoi il y a un message d'erreur :

```
>>> prix_initial = 20000
>>> prix_final = prxi_initial * 0.70
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
NameError: name 'prxi_initial' is not defined
```

Le nom `prxi_initial` n'est pas défini. C'est en fait `prix_initial` qu'il fallait écrire.

```
>>> def test (x, y):
    a = x + y
    b = x * y
    return a + b

File "<pyshell>", line 3
    b = x * y
IndentationError: unexpected indent
```

La première ligne de la fonction n'a pas la même indentation que les suivantes.

EXERCICE 7 : (1,5pt) On considère les deux fonctions ci-dessous :

```
def simple_au_double1(x):
    return x
    return 2*x
```

```
def simple_au_double2(x):
    print(x)
    print(2*x)
```

- 1) Indiquer les 2 expressions rentrées pour obtenir les résultats ci-dessous. Vous devez choisir parmi : `simple_au_double1(5)`, `simple_au_double2(5)`, `simple_au_double1(10)` ou `simple_au_double2(10)`

```
>>> simple_au_double2(10)
10
20
>>> simple_au_double1(10)
10
```

- 2) Entourer l'expression ci-dessous qui provoque une erreur :

- a) `1 + simple_au_double1(4)` b) `1 + simple_au_double2(4)`