Attention : certain programme donne des erreurs à l'exécution, vous devez l'indiquer par "Erreur".

Exercice 1:	(*)	
Zacroroc I .		>>> # script executed
	for i in range(3):	
	print("Bonjour ", i)	
Exercice 2:	(*)	
	a = 2+3*4	>>> # script executed
	print(a)	
Exercice 3:	(*)	>>> # script executed
	<pre>for k in range(2): print(k)</pre>	
Exercice 4:		>>> # script executed
	<pre>def carre(n): return n * n</pre>	* Script executed
	<pre>print(carre(5))</pre>	
	princ(carre(o))	
Exercice 5:	· ·	
	<pre>t = 'Berthelot' U = t.upper()</pre>	>>> #itt3
	n = 0	>>> # script executed
	for c in U: if c == 'T':	
	n = n+1	
	print(n)	
Exercice 6:	(*)	>>> # script executed
	t = 'Berthelot'	* Script executed
	<pre>U = t.upper() n = 0</pre>	
	for c in U:	
	if c == 'T': n = n+1	
	print(n)	

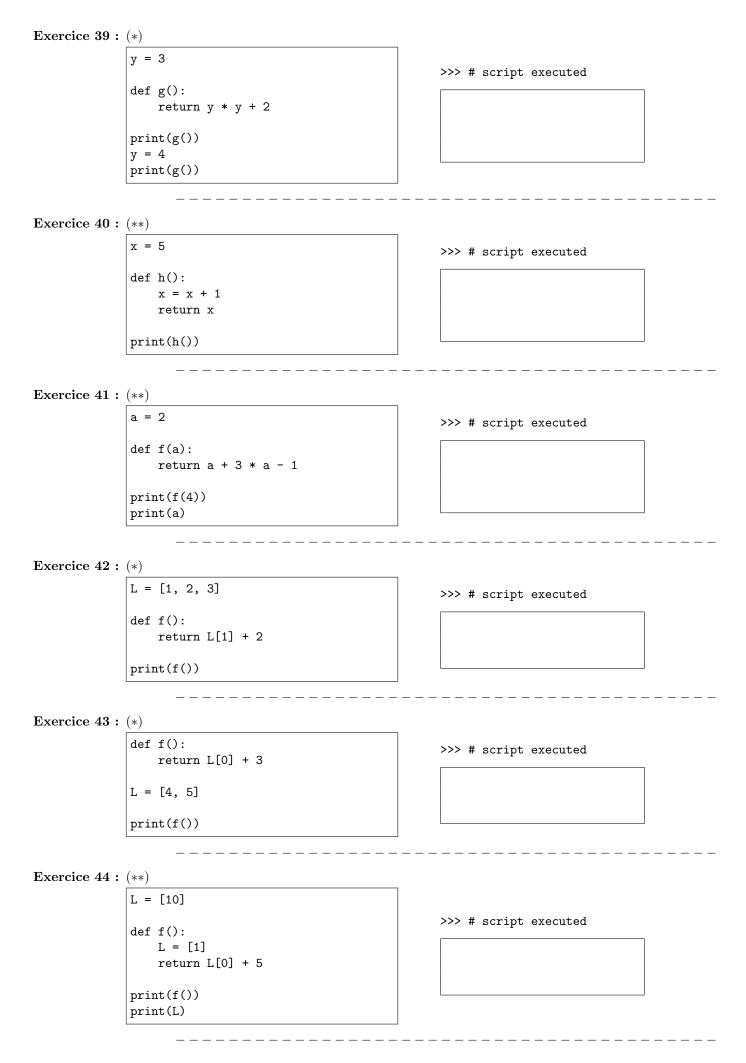
Exercice 7	: (*)	
	t = 'Berthelot'	
	<pre>U = t.upper()</pre>	>>> # script executed
	n = 0	
	for c in U:	
	if c == T:	
	n = n+1	
	<pre>print(n)</pre>	
	print(n)	
Exercice 8	• (4)	
Exercice 6		
	L = [1, 3]	>>> # script executed
	for x in L:	
	L.append(x)	
	<pre>print(L)</pre>	
Exercice 9	: (*)	
	a, b = 2, 3	
		>>> # script executed
	if a < b:	
	<pre>print("Affichage 1")</pre>	
	else:	
	<pre>print("Affichage 2")</pre>	
	1 . 0 .	
Exercice 10	: (*)	
	<pre>def f(x):</pre>	
	return x**x	>>> # garint avaguted
		>>> # script executed
	n = 1	
	while f(n) < 10:	
	n = n + 1	
	<pre>print(n)</pre>	
	•	
Exercice 11	: (*)	
		>>> # script executed
	1 0	
	k = 0	
	while k < 10:	
	k = k + 3	
	<pre>print(k)</pre>	
	P2 223 (22)	
Exercice 12	: (*)	
	L = [1, 2]	>>> # script executed
	while len(L) < 4:	
	L.append(0)	
	print(L)	
	br 1110 (11)	
Erroraia - 10	• (4)	
Exercice 13	: (*)	
	x = 5	>>> # script executed
	y = x	
	x = x + 1	
	<pre>print(y)</pre>	

Exercice 14:	(*)	
	<pre>def mystere(n):</pre>	
	if n % 2 == 0:	>>> # script executed
	return "Pair"	
	else:	
	return "Impair"	
	<pre>print(mystere(7))</pre>	
D • 15		
Exercice 15:	(*)	
	L = [0, 1, 2]	>>> # garint arequired
	for i in range(len(L)):	>>> # script executed
	L[i] = L[i] + 1	
	(7)	
	print(L)	
Exercice 16:	(*)	
		>>> # script executed
	s = "abcdef"	and a second sec
	<pre>print(s[3])</pre>	
Exercice 17:	(**)	
	n = 5	>>> #i+
	fact = 1	>>> # script executed
	while n > 1:	
	fact = fact * n	
	n = n - 1	
	<pre>print(fact)</pre>	
	P12110 (1 000)	
Exercice 18:	(**)	
	i = 0	
	s = 0	>>> # script executed
	while i < 4:	
	s = s + i	
	i = i + 1	
	print(s)	
Exercice 19:	(*)	
Exercice 19.	(*)	>>> # acript executed
		>>> # script executed
	a = 1	
	while a < 10:	
	print(a)	
	a = a * 2	
T • ~~	(*)	
Exercice 20:	- []	
Exercice 20:	L =	
Exercice 20:	L = []	>>> # script executed
Exercice 20:	x = 0	>>> # script executed
Exercice 20:	x = 0 while x < 3:	>>> # script executed
Exercice 20:	<pre>x = 0 while x < 3: L.append(x)</pre>	>>> # script executed
Exercice 20:	<pre>x = 0 while x < 3: L.append(x) x = x + 1</pre>	>>> # script executed
Exercice 20:	<pre>x = 0 while x < 3: L.append(x)</pre>	>>> # script executed

Exercice 21:	(*)	
	x = 10	>>> # script executed
	while x > 0:	
	if x % 3 == 0:	
	<pre>print("Divisible par 3 :", x)</pre>	
	x = x - 1	
Exercice 22:	(**)	
		>>> # script executed
	a = 2 ** 3 ** 2	
	print(a)	
Exercice 23:	(*)	
	x = 4	>>> # garint arrayted
	y = 3	>>> # script executed
	z = 2	
	r = x + y * z ** 2	
	print(r)	
	P1110(1)	
Exercice 24:	(*)	
Exercice 24.		>>> #i+
	a = 10	>>> # script executed
	b = 3	
	c = a / b * b	
	<pre>print(c == a)</pre>	
Exercice 25:	(**)	
	a = True	>>> # script executed
	b = False	
	c = a + b * 3	
	print(c)	
Exercice 26:	()	
Exercice 20:		
	a = 2	>>> # script executed
	b = 3	•
	c = 4	
	res = (a + b) * c // 2 ** 2	
	print(res)	
Exercice 27:	(*)	
	<pre>def f(x):</pre>	
	return x + 2	
	TOULH A · Z	>>> # script executed
	def g(x):	<u>-</u>
	return x * 3	
	160mm x + 2	
	$\frac{1}{2} = f(2) + g(1)$	
	a = f(2) + g(1)	
	print(a)	

Exercice 28:	(*)	
	<pre>def h(x, y): return x ** y</pre>	>>> # script executed
	res = h(2, 1 + 2) print(res)	
Exercice 29:		
	def f(a, b): return a + b * 2	>>> # script executed
	x = f(1 + 2, 3) print(x)	
Exercice 30:		
	<pre>def f(x): return x * x</pre>	
		>>> # script executed
	<pre>def g(x): return x + 1</pre>	
	res = f(g(2)) + g(f(2)) print(res)	
Exercice 31:	(*)	
	<pre>def min3(a, b, c): return min(a + b, c)</pre>	>>> # script executed
	x = min3(2, 3 * 2, 10) print(x)	
Exercice 32:	(*)	
	x = 10	>>> # script executed
	<pre>def f(): x = 5</pre>	
	return x + 2	
	<pre>print(f()) print(x)</pre>	
Exercice 33:	(***)	
	a = 2	>>> # script executed
	def g():	
	a = a + 1 return a	
	<pre>print(g())</pre>	

Exercice $34:$	(**)	
	b = 4	>>> # script executed
	def h(b): return b * 2 + 1	
	<pre>print(h(3)) print(b)</pre>	
Exercice 35:	(*)	
	y = 1	>>> # script executed
	<pre>def f(x): y = x * 2 return y + x</pre>	
	<pre>print(f(3)) print(y)</pre>	
Exercice 36:	(*)	
Exercise 50.	m = 1	
	<pre>def f(): m = 2</pre>	>>> # script executed
	return m ** 2	
	<pre>def g(): return m + 3</pre>	
	<pre>print(f() + g())</pre>	
T		
Exercice 37:	(*) def f():	
	return x + 2	>>> # script executed
	x = 10	
	<pre>print(f()) print(x)</pre>	
Exercice 38:		
	x = 10	>>> # garint everyted
	<pre>def f(): x = 5 return x * 2 + 1</pre>	>>> # script executed
	<pre>print(f()) print(x)</pre>	



Exercice 45:	(**)	
	L = [2, 4]	
		>>> # script executed
	def f():	
	L[0] = L[0] + 3	
	return L	
	<pre>print(f())</pre>	
	print(L)	
Exercice 46:	(**)	
	L = [1, 2]	
		>>> # script executed
	<pre>def f():</pre>	
	L.append(3)	
	f() print(L)	
	print(L)	
Exercice 47:	(**)	
Exercise II.		
	<pre>def f(x): return x**n</pre>	
	recurn x**n	>>> # script executed
	L = []	
	for n in range(3):	
	L.append(f(2))	
	print(L)	
Exercice 48:	(**)	
	L = [1]	>>> # script executed
	<pre>def f(): L = L + [2]</pre>	
	return L	
	<pre>print(f())</pre>	
Exercice 49:	(*)	
	L = []	>>> # script executed
	for k in range(11):	
	L.append(k/10)	
	print(L)	
Eventing 50 a (v)		
Exercice 50:		
	<pre>def f(x):</pre>	>>> #it
	return x**2	>>> # script executed
	f = f(3)	
	1 - 1(3) print(f)	
	f = f(4)	
	print(f)	

a - True	Exercice 51:	(*)	
Exercice 52: (*) x = 5		a = True	>>> # script executed
Exercise 52: (*) x = 5 y = 0 print((x > 0) and (y != 0)) Exercise 53: (*) a = True b - False print(not (a or b)) Exercise 54: (*) x = 3 y = 7 z = 10 print((x < y) or (z < y) and not (x == 3)) Exercise 55: (*) val = 0 if not val: print("Condition vraie") else: print("Condition fausse") Exercise 56: (*) a = True b - True c - False print("Cand b) or (not c)) Exercise 57: (*) s = "abc" t = "de" print(s + t) Exercise 58: (*) Exercise 58: (*) s = "ha" print(s + 3) Exercise 59: (*) s = "Python" Exercise 59: (*) s = "Python" >>> # script executed ** ** ** ** ** ** ** ** ** ** ** ** *			
Exercise 52: (*) x = 5 y = 0 print((x > 0) and (y != 0)) Exercise 53: (*) a = True b - False print(not (a or b)) Exercise 54: (*) x = 3 y = 7 z = 10 print((x < y) or (z < y) and not (x == 3)) Exercise 55: (*) val = 0 if not val: print("Condition vraie") else: print("Condition fausse") Exercise 56: (*) a = True b - True c - False print("Cand b) or (not c)) Exercise 57: (*) s = "abc" t = "de" print(s + t) Exercise 58: (*) Exercise 58: (*) s = "ha" print(s + 3) Exercise 59: (*) s = "Python" Exercise 59: (*) s = "Python" >>> # script executed ** ** ** ** ** ** ** ** ** ** ** ** *		print(a and b)	
X = 5			
X = 5			
X = 5	Exercice 52:	(*)	
Exercice 53: (*) a = True b = False print((x > 0) and (y != 0)) Exercice 54: (*) x = 3 y = 7 z = 10 print((x < y) or (z < y) and not (x == 3)) Exercice 55: (*) val = 0 if not val: print("Condition vraie") elase: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print("and b) or (not c)) Exercice 57: (*) Exercice 58: (*) Exercice 58: (*) Exercice 58: (*) Exercice 59: (*)	<u> </u>		XXX # gamint arrespeted
Exercice 53: (*) a = True			/// # Script executed
Exercice 53 : (*) a = True b = False print(not (a or b)) Exercice 54 : (*) x = 3 y = 7 z = 10 print((x < y) or (z < y) and not (x == 3)) Exercice 55 : (*) val = 0 if not val: print("Condition vraie") else: print("Condition fausse") Exercice 56 : (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "da" print(s + t) Exercice 58 : (*) Exercice 58 : (*) Exercice 59 : (*) S = "Python" **Seript executed		1 7	
a = True		print((x > 0)) and $(y != 0))$	
a = True			
a = True			
b = False	Exercice 53:	(*)	
Exercice 54: (*) x = 3		a = True	>>> # script executed
Exercice 54: (*) x = 3 y = 7 z = 10 print((x < y) or (z < y) and not (x == 3)) Exercice 55: (*) Val = 0 if not val: print("Condition vraie") else: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s + 3) Exercice 59: (*) s = "Python" >>> # script executed		b = False	
Exercice 54: (*) x = 3 y = 7 z = 10 print((x < y) or (z < y) and not (x == 3)) Exercice 55: (*) Val = 0 if not val: print("Condition vraie") else: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s + 3) Exercice 59: (*) s = "Python" >>> # script executed		<pre>print(not (a or b))</pre>	
X = 3			
X = 3			
X = 3	Exercice 54 ·	(*)	
y = 7	Exercise 91.		
<pre>Exercice 55: (*) Exercice 55: (*) val = 0 if not val: print("Condition vraie") else: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) Exercice 59: (*) s = "ha" print(s * 3)</pre> Exercice 59: (*) Exercice 59: (*) s = "Python" >>> # script executed Exercice 59: (*) S = "Python" >>> # script executed >>> # script executed			>>> # script executed
Exercice 55: (*) val = 0		1 -	
Exercice 55: (*) val = 0			
val = 0 if not val:		print((x < y) or (z < y) and not (x == 3))	
val = 0 if not val:			
val = 0 if not val:			
if not val: print("Condition vraie") else: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed	Exercice 55:	(*)	
if not val: print("Condition vraie") else: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed		val = 0	
<pre>print("Condition vraie") else: print("Condition fausse") Exercice 56 : (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57 : (*) s = "abc" t = "de" print(s + t) Exercice 58 : (*) s = "ha" print(s * 3) Exercice 59 : (*) s = "Python"</pre> >>> # script executed # script executed # scr			>>> # script executed
<pre>else: print("Condition fausse") Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed</pre>			
Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed			
Exercice 56: (*) a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed		<pre>print("Condition fausse")</pre>	
<pre>a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python"</pre> >>> # script executed		_	
<pre>a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python"</pre> >>> # script executed			
<pre>a = True b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python"</pre> >>> # script executed	Exercice 56:	(*)	
b = True c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed	Exercise 50.		
<pre>c = False print((a and b) or (not c)) Exercice 57: (*) s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python"</pre> >>> # script executed			>>> # script executed
Exercice 57: (*) s = "abc"			
Exercice 57: (*) S = "abc"			
<pre>s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed</pre>		print((a and b) or (not c))	
<pre>s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed</pre>			
<pre>s = "abc" t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed</pre>			
<pre>t = "de" print(s + t) Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*)</pre>	Exercice 57:	(*)	
<pre>t = "de" print(s + t) Exercice 58: (*)</pre>		s = "abc"	>>> # script executed
Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed			
Exercice 58: (*) s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed		<pre>print(s + t)</pre>	
<pre>s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed >>> # script executed</pre>			
<pre>s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed >>> # script executed</pre>			
<pre>s = "ha" print(s * 3) Exercice 59: (*) s = "Python" >>> # script executed >>> # script executed >>> # script executed</pre>	Exercice 58 ·	(*)	
<pre>s = "ha" print(s * 3) Exercice 59: (*)</pre>	Exercise 50.		>>> # script executed
Exercice 59: (*) s = "Python" >>> # script executed			
s = "Python" >>> # script executed		<pre>print(s * 3)</pre>	
s = "Python" >>> # script executed			
s = "Python" >>> # script executed			
s = "Python" >>> # script executed	Exercice 50 ·	(*)	
s = "Python"	DACICICE 00:		>>> # script executed
print(s[0], s[-1])			232_F3 38334304
		print(s[0], s[-1])	

Exercice 60:	(*)	
	s = "Bonjour"	>>> # script executed
	print(len(s))	
	print(ich(s))	
T		
Exercice 61:	(**)	
	def f2(mot):	
	v = "aeiouy"	>>> # script executed
	for c in mot.lower():	** * Bolipu executed
	if c in v:	
	return True	
	return False	
	<pre>print(f2("Anne"))</pre>	
	print(f2("SNCF"))	
Exercice 62:	(*)	
	def f3(mot):	
	if len(mot) > 5:	>>> # script executed
	return mot.upper()	
	else:	
	return mot.lower()	
	<pre>print(f3("python"))</pre>	
	<pre>print(f3("code"))</pre>	
Exercice 63:	(**)	
	def f5(s1, s2):	>>> # script executed
	return len(s1) == len(s2)	-
	<pre>print(f5("chat", "chien"))</pre>	
	print(f5("lune", "mars"))	
	P-110(10(1010)	
Exercice 64:	(++)	
Exercice 04.		
	def f1(a, b):	>>> # script executed
	return (a > 0) and (b > 0)	
	print(f1(3, 5))	
	print(f1(-2, 7))	
Exercice 65:	(**)	
	def f3(a, b, c):	>>> # script executed
	return (a == b) and not (b == c)	
	return (a b) and not (b c)	
	print(f3(2, 2, 3))	
	print(f3(2, 2, 3)) print(f3(4, 4, 4))	
	print(13(4, 4, 4))	
E	(.)	
Exercice 66:	(*)	
	def f5(n):	>>> # script executed
	return not (n < 10 or n > 20)	
	print(f5(15))	
	print(f5(25))	
	_	