1. What is the output of the following function call

Quelle est la sortie de l'appel de fonction suivant

```
def outerFun(a, b):
    def innerFun(c, d):
        return c + d
    return innerFun(a, b)
    return a


result = outerFun(5, 10)
print(result)
```

1. 5
2. 15
3. (15, 5)
4. Syntax Error

2. What is the output of the add() function call

Quelle est la sortie de l'appel de fonction add ()

```
def add(a, b):
    return a+5, b+5

result = add(3, 2)
print(result)
```

1. 15
2. 8
3. (8, 7)
4. Syntax Error

3. Choose the correct function declaration of fun1() so that we can execute the following function call successfully

Choisissez la déclaration de fonction correcte de fun1 () afin que nous puissions exécuter l'appel de fonction suivant avec succès

```
fun1(25, 75, 55)

fun1(10, 20)
```

1. def fun1(**kwargs)
2. No, it is not possible in Python
3. def fun1(args*)
4. def fun1(*data)

Le *packing (paquete)* et l'*unpacking (depaquete)* sont deux concepts Python complémentaires qui offrent la possibilité de transformer une succession d'arguments, nommés ou non, en liste/tuple ou dictionnaire et vice-versa.

4. What is the output of the following function call

Quelle est la sortie de l'appel de fonction suivant

```
def fun1(name, age=20):
    print(name, age)


fun1('Emma', 25)
```

1. Emma 25
2. Emma 20

5. What is the output of the following code

Quelle est la sortie du code suivant

```
def outerFun(a, b):
    def innerFun(c, d):
        return c + d
    return innerFun(a, b)

res = outerFun(5, 10)
print(res)
```

1. 15
2. Syntax Error
3. (5, 10)

6. Select which true for Python function

Sélectionnez le vrai pour la fonction Python

1. A function is a code block that only executes when it is called.
2. Python function always returns a value.
3. A function only executes when it is called and we can reuse it in a program
4. Python doesn’t support nested function

7. What is the output of the following displayPerson() function call

Quelle est la sortie de l'appel de fonction displayPerson () suivant

```
def displayPerson(*args):
    for i in args:
        print(i)


displayPerson(name="Emma", age="25")
```

1. TypeError
2. Emma
   25
3. name
   age

8. Python function always returns a value

La fonction Python renvoie toujours une valeur

1. False
2. True

9. What is the output of the following display() function call

Quelle est la sortie de l'appel de fonction display () suivant

```
def display(**kwargs):
    for i in kwargs:
        print(i)
display(emp="Kelly", salary=9000)
```

1. TypeError
2. Kelly
   9000
3. (’emp’, ‘Kelly’)
   (‘salary’, 9000)
4. emp
   salary

10. What is the output of the following function call

Quelle est la sortie de l'appel de fonction suivant

```
def fun1(num):
    return num + 25

fun1(5)
print(num)
```

1. 25
2. 5
3. NameError

11. Select which is true for Python function

Sélectionnez ce qui est vrai pour la fonction Python

1. A Python function can return only a single value
2. A function can take an unlimited number of arguments.
3. A Python function can return multiple values
4. Python function doesn’t return anything unless and until you add a return statement

12. Given the following function fun1() Please select the correct function calls

Étant donné la fonction suivante fun1 () Veuillez sélectionner les appels de fonction corrects

```
def fun1(name, age):
    print(name, age)
```

1. fun1(name=’Emma’, age=23)
2. fun1(name=’Emma’, 23)
3. fun1(‘Emma’, 23)

13. Which of the following lines properly starts a parameter less function definition?

Laquelle des lignes suivantes démarre correctement une définition de fonction sans paramètre?

1. **function fun():**
2. **def fun:**
3. **fun function():**
4. **def fun():**

14. A function defined in the following way:

Une fonction définie de la manière suivante:

```
def function(x=0):
    return x
```

1. may be invoked without any argument, or with just one
2. must be invoked without arguments
3. must be invoked with exactly one argument
4. may be invoked with any number of arguments (including zero)

15. A built-in function is a function which:

Une fonction intégrée est une fonction qui:

1. **has to be imported before use**
2. **is hidden from programmers**
3. **has been placed within your code by another programmer**
4. **comes with Python, and is an integral part of Python**

16. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def f(x):
    if x == 0:
        return 0
    return x + f(x – 1)
print(f(3))
```

1. the code is erroneous
2. 3
3. 1
4. 6

17. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def fun(x):
    x += 1
    return x
x = 2
x = fun(x+1)
print(x)
```

1. the code is erroneous
2. 4
3. 5
4. 3

18. The following snippet:

L'extrait suivant:

```
def func(a,b):
return a ** a
print(func(2))
```

1. will output 2
2. is erroneous
3. will return None
4. will output 4

19. The following snippet:

L'extrait suivant:

```
def func1(a):
    return a ** a
def func2(a):
    return func1(a)*func1(a)
print(func2(2))
```

1. is erroneous
2. will output 2
3. will output 4
4. will output 16

20. Which of the following lines properly starts a function using two parameters, both with zeroed default values?

Laquelle des lignes suivantes démarre correctement une fonction en utilisant deux paramètres, tous deux avec des valeurs par défaut mises à zéro?

1. **def fun(a=b=0):**
2. **fun fun(a,b=0):**
3. **fun fun(a=0,b):**
4. **def fun(a=0,b=0):**

21. Which of the following statements is false?

Lequel des énoncés suivants est faux?

1. **The None value cannot be used as an argument of arithmetic operators**
2. **The None value can be assigned to variables**
3. **The None value may not be used outside functions**
4. **The None value can be compared with variables**

22. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def fun(x):
    if x % 2 == 0:
        return 1
    else:
        return
print(fun(fun(2)) + 1)
```

1. the code will cause a runtime error
2. 1
3. None
4. 2

23. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def fun(x):
    global y
    y = x * x
    return y
fun(2)
print(y)
```

1. 2
2. 4
3. None
4. the code will cause a runtime error

24. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def any():
    print(var + 1,end=‘’)


var = 1
any()
print(var)
```

1. 22
2. 21
3. 11
4. 12

25. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def fun(x,y,z):
    return x+2*y+3*z
print(fun(0,z=1,y=3))
```

1. 3
2. the snippet is erroneous
3. 0
4. 9

26. What is the output of the following snippet?

Quelle est la sortie de l'extrait de code suivant?

```
def fun(inp=2,out=3):
    return inp * out
print(fun(out=2))
```

1. 6
2. 4
3. 2
4. the snippet is erroneous