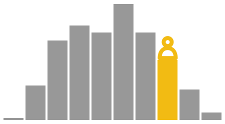
Afin d'éviter la fuite de questions, ce rapport ne doit pas être partagé avec les candidats.

Anis HAMROUN (anishamroun2@gmail.com)

Campagne : MERITIS - Python **Langage(s) de programmation :** Python3 **Langage :** Anglais **Date :** 11/06/2019

SCORE	RANG	DURÉE	MEILLEUR QUE
72% 1 882 / 2 620 pts	24 / 79	0H55 / 1H53	86% des développeurs



Python3

72%
(1 882 / 2 620)

Connaissance du langage	56%	(441 / 781)
Fiabilité	91%	(481 / 526)
Modélisation	87%	(260 / 300)
Résolution de problèmes	69%	(700 / 1 013)

CodinGame FOR WORK

Question 1: import os

Python3 00:32 / 00:35 20 / 20 pts

Question

You are about to write Python code to manage files in a given directory.
Which module(s) will you import to programmatically navigate the file system?

Réponse

● import os
○ import fs
○ import inspect
○ import filesystem

Résultat

✓ Réponse correcte
Connaissance du langage +20pts

Question 2: Boolean expression

Python3 00:29 / 00:30 20 / 20 pts

Question

What would you use to apply an AND operation between two boolean values in Python?

Réponse

○ &&

● *and*

○ .

○ ||

Résultat

✓ Réponse correcte
Connaissance du langage +20pts

Question 3: Dict definition

Python3 00:15 / 00:30 20 / 20 pts

Question

```
val = { '0' : 0, '1' : 1 , '2' : 2}
```

What is the type of ***val***?

Réponse

● *dict*
○ *list*
○ *array*
○ *map*
○ *set*

Résultat

✓ Réponse correcte
Connaissance du langage +20pts

Question 4: Object instantiation

Python3 00:22 / 00:45 0 / 20 pts

Question

How does one create a new instance ***point*** of the following object:

```
class Point():
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __eq__(self, other):
        return (self.x, self.y) == (other.x, other.y)
```

Réponse

● *point = new Point(x, y)*

○ *point = Point(point, x, y)*

○ *point = Point(x, y)*

Résultat

⊗ Réponse incorrecte
Connaissance du langage ~~+20pts~~

Question 5: Threads

Python3 00:08 / 00:25 20 / 20 pts

Question

A Python program can be multi-threaded

Réponse

● True

○ False

Résultat

✓ Réponse correcte
Modélisation +20pts

Question 6: Adding an element to a set

Python3 00:36 / 00:40 40 / 40 pts

Question

Which of these instructions can be used to add the number 5 to a set named ***values***?

Check all valid answers.

Réponse

○ *values.append(5)*

● *values.add(5)*

○ *values += 5*

Résultat

✓ Réponse correcte

Connaissance du langage +40pts

Question 7: Raising an exception

Python3 00:30 / 00:30 0 / 40 pts

Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

Question

In Python3, which of these instructions can be used to create an exception?

Check all valid answers.

Réponse

☐ raise Exception()
☐ throw Exception()
☐ raise Exception("Exception occurred!")
☑ throw Exception("Exception occurred!")

Résultat

Réponse incorrecte
Connaissance du langage ~~+40pts~~

Question 8: Existence of key in a dict

Python3 00:59 / 01:00 40 / 40 pts

Question

Which of these instructions can you use to check if the key ***"Bob"*** is present in the ***phonebook*** dictionary?

Réponse

● "Bob" in phonebook

○ phonebook["Bob"] is not None

○ phonebook["Bob"] != None

○ phonebook.Bob != None

○ phonebook.contains("Bob")

Résultat

✓ Réponse correcte
Connaissance du langage +40pts

Question 9: Execution order

Python3 00:41 / 01:00 40 / 40 pts

Question

The code below is in a file called ***file.py***. If you were to execute the commmand ***python3 file.py***, in what order would the code blocks be executed?

```
#code block A

def main():
    #code block B

if __name__ == '__main__':
    main()

#code block C
```

Réponse

● A then B then C
○ only B is executed
○ A then B
○ A then C then B
○ A then C

Résultat

✓ Réponse correcte
Connaissance du langage +40pts

Question 10: Function declaration

Python3 00:07 / 00:30 20 / 20 pts

Question

How does one declare a function in Python?

Check all valid answers.

Réponse

◉ *def name():*

○ *name():*

○ *function name():*

○ *void name():*

○ There are no functions in Python

Résultat

✓ Réponse correcte

Connaissance du langage +20pts

Question 11: Types of tuple elements

Python3 00:20 / 00:30 20 / 20 pts

Question

Can the elements of a ***tuple*** be of different types?

Réponse

● Yes

○ No

Résultat

✓ Réponse correcte
Connaissance du langage +20pts

Question 12: Indexation

Python3 00:11 / 00:35 0 / 60 pts

Question

Which of these instructions can you use to get the last element of the following list: ***arr*** = *[1,2,3,4,5]*?

Check all valid answers.

Réponse

☑ *arr[-1]*

□ *arr[len(arr)-1]*

□ *arr.last()*

□ *reversed(arr)[0]*

□ *last(arr)*

Résultat

☒ Réponse incorrecte

Connaissance du langage ~~+60pts~~

Question 13: Iterate over a string

Python3 00:35 / 00:35 0 / 60 pts

Question

Which of these instructions can be used to iterate over the characters of the string ***string***?

Check all valid answers.

Réponse

- ☑ *for c in string:*
- ☑ *for c in string.split(''):*
- ☑ *for c in list(string):*
- □ *with string as c:*

Résultat

☒ Réponse incorrecte
Connaissance du langage ~~+60pts~~

Question 14: Tuple vs List

Python3 00:20 / 01:00 0 / 40 pts

Question

What is the difference between a ***tuple*** and a ***list***?

Réponse

○ ***list*** is ordered, ***tuple*** is unordered.

○ ***list*** can contain duplicates, ***tuple*** contains no duplicates.

● ***tuple*** may contain values of different types, ***list*** may not.

○ ***tuple* is immutable, *list* is mutable.**

Résultat

☒ Réponse incorrecte
Connaissance du langage ~~+40pts~~

Question 15: super()

Python3 00:39 / 00:45 0 / 40 pts

Question

Consider the Python 3 code below.

```
class A:
    def __init__(self, text):
        self.text = text

class B(A):
    def __init__(self):
        #TODO
```

Among these instructions, which one should be used to replace the #TODO?

Réponse

○ `super().__init__('hello')`

● `super('hello')`

Résultat

⊗ Réponse incorrecte

Modélisation ~~+40pts~~

Question 16: Sort

Python3 00:19 / 00:35 0 / 60 pts

Question

Which of these instructions let you sort the list

```
arr = [0,3,1,2,4]
```

in ascending order?

Check all valid answers.

Réponse

- □ *arr.sort()*
- □ *sort(arr)*
- ☑ *arr = sorted(arr)*
- □ *arr[::1]*
- □ None of the above

Résultat

⊗ Réponse incorrecte
Connaissance du langage ~~+60pts~~

Question 17: Mapping

Python3 00:29 / 00:45 0 / 60 pts

Question

Which of the following instructions let you transform a list of strings

```
strs = ['0', '1', '2']
```

into a list of integers

```
[0,1,2]
```

?

Check all valid answers.

Réponse

☐ *list(map(int, strs))*

☑ *[int(x) for x in strs]*

☑ *strs.map(lambda x: int(x))*

☐ None of the above

Résultat

☒ Réponse incorrecte
Connaissance du langage ~~+60pts~~

Question 18: Strings equality

Python3 00:38 / 02:30 50 / 50 pts

Question

is_foo(param) should return ***True*** if ***param*** is equal to the string ***"foo"***, ***False*** otherwise.

Implement ***is_foo(param)*** function.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def is_foo(param):
    # Your code goes here
    return param == "foo"
```

Résultat

✓ Use of the operator '=='
Résolution de problèmes +17pts

✓ Still works if param is None
Fiabilité +33pts

Question 19: Inheritance

Python3 01:26 / 02:00 50 / 50 pts

Question

Complete the answer to make B inherit from A.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

class A():
    def test(self):
        print("A")

    def __init__(self):
        self.test()


class B(A):
    def __init__(self):
        self.test()
```

Résultat

✓ B inherited from A
Connaissance du langage +50pts

Question 20: Correction

Python3 00:43 / 05:00 100 / 100 pts

Question

The following *factorial* function written by your colleague Fred is supposed to return the factorial of a number, but it is broken.

Fix the *factorial* function.

As a reminder: ***factorial(n)* = *1* * *2* * *3* * *...* * *n***

Réponse

```
# Python code below
# Use print("messages...") to debug your solution.

def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

Résultat

✓ The factorial function works again
Connaissance du langage +100pts

Question 21: Largest wins from chaos

Python3 00:25 / 05:00 100 / 100 pts

Question

find_ largest(numbers) should return the largest number from ***numbers***. The array ***numbers*** always contains at least one number.

Implement ***find_ largest(numbers)***.

Réponse

```
# Python code below
# Use print("messages...") to debug your solution.

def find_largest(numbers):
    # Your code goes here
    return max(numbers)
```

Résultat

✓ It works using simple data sample
Résolution de problèmes +32pts

✓ Still works when the array contains only Integer.MIN_VALUE
Fiabilité +58pts

✓ Still works if the largest number is at position 0 in the array
Fiabilité +5pts

✓ Still works if the largest number is at the last position in the array
Fiabilité +5pts

Question 22: Average

Python3 00:45 / 02:30 75 / 100 pts

Question

Write the body of the ***average(table)*** function.

The function should return the average of the values contained in ***table***. ***table*** is always a defined array, objects in ***table*** are always numbers.

average should return ***0*** if ***table*** is empty.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def average(table):
    # Your code goes here
    return sum(table) / len(table)
```

Résultat

✓ The solutions computes averages
Connaissance du langage +50pts

✗ The solution works for an empty set
Fiabilité ~~+25pts~~

✓ The solution works for a large set
Fiabilité +25pts

Question 23: Simple boolean expression

Python3 01:20 / 02:00 100 / 100 pts

Question

is_bool(i, j) should return ***True*** if one of the arguments equals 1 or if their sum is equal to 1.

For example:
is_bool*(1, 5)* returns ***True***
is_bool*(2, 3)* returns ***False***
is_bool*(-3, 4)* returns ***True***

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def is_bool(i, j):
    # Your code goes here

    return  i == 1 or j == 1 or (i + j) == 1
```

Résultat

✓ Returns True if i or j equals 1, False otherwise
Résolution de problèmes +67pts

✓ Returns True if i+j equals 1
Fiabilité +33pts

Question 24: Twins

Python3 02:50 / 15:00 150 / 150 pts

Question

A twin of a word is a word written with the same letters (case insensitive) but not necessarily in the same order.

For example ***Silent*** is a twin of ***Listen***.

The ***is_twin(a, b)*** function should return ***True*** if ***b*** is the twin of ***a*** and ***False*** otherwise. a and b are two strings and are not None.

Write the body of the ***is_twin(a, b)*** function.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def is_twin(a, b):
    # Your code goes here

    return sorted(a.lower()) == sorted(b.lower())
```

Résultat

✓ Simple tests
Résolution de problèmes +55pts

✓ Case insensitive
Résolution de problèmes +30pts

✓ Empty and one letter strings
Fiabilité +20pts

✓ Same letters but no twins
Fiabilité +45pts

Question 25: Array Index

Python3 02:34 / 03:00 67 / 100 pts

Question

Write the body of the function ***is_on_even_position*(table, value)**. The function should return ***True*** if ***value*** is contained in ***table*** at an even index, ***False*** otherwise.

Note: the table parameter is never None.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def is_on_even_position(table, value):
    # Your code goes here

    return table.index(value) % 2 == 0
```

Résultat

✓ The solution works for standard cases
Résolution de problèmes +40pts

✗ The solution works with an empty table
Fiabilité ~~+7pts~~

✗ Value is at an even and odd position
Fiabilité ~~+13pts~~

✓ The solution works on a large table
Fiabilité +13pts

✗ The solution works when value is not in table
Résolution de problèmes ~~+13pts~~

✓ The solution works for boundaries
Résolution de problèmes +14pts

Question 26: Approximation of π

Python3 05:56 / 12:00 200 / 200 pts

Question

In this exercise we will calculate an approximation of π (Pi).

The technique is as follows:
Take a random point P at coordinated (x, y) such that $0 \le x \le 1$ and $0 \le y \le 1$. If $x^2 + y^2 \le 1$, then the point is inside the quarter disk of radius 1, otherwise the point is outside.

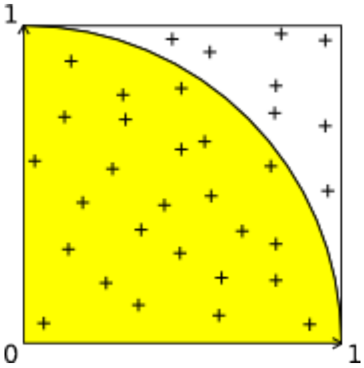


Fig 1. An example using 33 random points.

We know that the probability that the point is inside the quarter disk is equal to $\pi/4$.

Write the ***piApprox(pts)*** function who will use the points ***pts*** to return an approximation of the number float π.
pts is a multidimentional list of float.

Input:
Each item in ***pts*** is a point. A point is represented by an array containing exactly two numbers, respectively, x and y such that $0 \le x \le 1$ and $0 \le y \le 1$. ***pts*** is never None and always contains at least one item.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def pi_approx(pts):
    # Your code goes here

    nb_pts = len(pts)
    nb_included_pts = 0

    for (x,y) in pts:
        if( x**2 + y**2 <= 1):
            nb_included_pts += 1


    return 4 * nb_included_pts /nb_pts
```

Résultat

✓ Approximation of π is correct (related to pts)
Résolution de problèmes +171pts

✓ The point P(1, 0) is inside the quarter disk
Fiabilité +29pts

Question 27: ASCII Art

Python3 04:13 / 10:00 300 / 300 pts

Question

print_char displays exactly one ASCII character from A to Z (inclusive) on multiple rows and columns.

Now, we want to do the reverse operation: to get a character from its graphic representation.

Implement the function ***scan_char(s)*** so that it returns the character associated with the graphical representation provided by ***print_char(c)***. If ***s*** does not match a character from A to Z (inclusive), then ***scan_char*** must return the character ***?***.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.
def scan_char(s):

    alphabet = "AZERTYUIOPQSDFGHJKLMWXCVBN"
    for letter in alphabet:
        if(s == print_char(letter)):
            return letter

    return '?'
```

Résultat

✓ The solution returns 'A' if, and only if, the string corresponds to A.
Résolution de problèmes +60pts

✓ The solution works from 'A' to 'Y'
Modélisation +180pts

✓ The solution returns '?' if no character matches s
Modélisation +30pts

✓ The solution works with 'Z'
Modélisation +30pts

Question 28: Reshape String

Python3 11:45 / 12:00 0 / 300 pts

Question

The ***reshape(n, str)*** function should return the string ***str*** without spaces and layed out in lines of at most ***n*** characters.

Examples:

```
reshape(3, "abc de fghij") => returns "abc\ndef\nghi\nj"
reshape(6, "1 23 456") => returns "123456"
```

Write the body of the reshape(n, str) function.

Réponse

```
# Python code below
# Use print("messages...") to debug your solution.

def reshape(n, str):
    # Your code goes here

    new_str = str.replace(" ", "")

    list_elements = ""

    sub_list_elements = []
    for letter in new_str :
        sub_list_elements += [ letter ]
        if(  len(sub_list_elements) >= n ):
            list_elements += "\n" + "".join([sub_list_elements])
            sub_list_elements = []

    list_elements += "\n" + "".join([sub_list_elements])
    #print(list_elements)
    return list_elements,str
```

Résultat

Basic tests
Résolution de problèmes ~~+188pts~~

NA Spaces and unique line management
Résolution de problèmes ~~+112pts~~

Question 29: Move towards zero

Python3 05:00 / 15:00 150 / 150 pts

Question

Implement ***closest_to_zero*** function to return the integer in the array ***ints*** that is closest to zero. If there are two integers equally close to zero, consider the positive element to be closer to zero (example: if ***ints*** contains -5 and 5, return 5). If ***ints*** is ***None*** or empty, return 0.

Input: integers in ***ints*** have values ranging from -2147483647 to 2147483647.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def closest_to_zero(ints):
    # Your code goes here

    if( ints == None or len(ints) == 0 ):
        return 0
    closest_elem = ints[0]

    for elem in ints:
        if( abs(elem) < abs(closest_elem) or ( elem > 0 and elem == -1 * closest_elem  ) ):
            closest_elem = elem

    return closest_elem
```

Résultat

✓ The result is correct with a simple data set [7, 5, 9, 1, 4]
Résolution de problèmes +64pts

- ✓ The solution works with 2147483647 or -2147483647
  Fiabilité +10pts

- ✓ The solution works when the array contains only negative integers
  Fiabilité +11pts

- ✓ When two integers are as close to 0, then the positive wins
  Fiabilité +11pts

- ✓ The solution works when the array contains only two equal negative integers
  Fiabilité +11pts

- ✓ The solution uses java.lang.Math.abs()
  Connaissance du langage +21pts

- ✓ The solution works with an empty array
  Fiabilité +11pts

- ✓ The solution works with a null array
  Fiabilité +11pts

Question 30: Combination options in a tournament

Python3 09:07 / 15:00 300 / 300 pts

Question

You have to organize a chess tournament in which players will compete head-to-head.

Here is how we proceed to form the duels: select a first player randomly, then, select his opponent at random among the remaining participants. The pair of competitors obtained forms one of the duels of the tournament. We proceed in the same manner to form all the other pairs.

In this exercise, we would like to know how many different pairs it is possible to form knowing that the order of opponents in a pair does not matter.

For example, with 4 players named A, B, C and D, it is possible to get 6 different pairs : AB, AC, AD, BC, BD, CD.

Implement ***count*** to return the number of possible pairs. Parameter ***n*** corresponds to the number of players.

Try to optimize your solution so that, ideally, the duration of treatment is the same for any ***n***.

Input: ***2 <= n <= 10000***

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def count(n):
    # Your code goes here

    return n * (n-1) / 2
```

Résultat

✓ The solution works with a simple data set
Résolution de problèmes +150pts

✓ The solution works with data close to 10000
Fiabilité +100pts

✓ The solution returns the correct result in constant time
Fiabilité +50pts

Glossaire

Connaissance du langage

La mesure de cette compétence permet de déterminer l'expérience du candidat dans la pratique d'un langage de programmation. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui devra être rapidement opérationnel.**

Modélisation

Cette mesure fournit une indication sur la capacité du candidat à appliquer des solutions standard pour résoudre des problèmes récurrents. Un développeur ayant un bon niveau dans cette compétence augmentera la qualité (maintenabilité, évolutivité) de vos applications. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui sera amené à travailler sur les briques qui structurent vos applications, à anticiper les besoins de demain pour développer des solutions pérennes.**

Résolution de problèmes

Cette compétence correspond aux aptitudes du candidat à comprendre et à structurer son raisonnement pour trouver des solutions à des problèmes complexes. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vos applications ont une composante technique importante (R&D, innovation).**

Fiabilité

La fiabilité caractérise la capacité du candidat à réaliser des solutions qui prennent en compte les cas particuliers. Plus cette compétence est élevée, plus vos applications sont robustes (moins de bugs).