Contenido creado por: Isabel Maniega

Question 1 (Basics)

You have the following file.

index.py:

```
from sys import argv
sum = 0
for i in range(2, len(argv)):
    sum += float(argv[i])
print(
    "The average score for {0} is {1:.2f}"
    .format(argv[1], sum/(len(argv)-2))
)
```

You want the following output.

```
The average score for Peter is 200.00
```

Which command do you have to execute in the command line?

A. `python index.py Peter 100 200 300`

B. `python index.py Peter 100`

C. `python index.py 100 200`

D. The code is erroneous

Solution 1

In [2]:

```
# index.py:
"""
from sys import argv
sum = 0
for i in range(2, len(argv)):
    sum += float(argv[i])
print(
    "The average score for {0} is {1:.2f}"
    .format(argv[1], sum/(len(argv)-2))
)

"""
# You want the following output.
# The average score for Peter is 200.00
# index3 en la carpeta prueba
# python3 index3.py Peter 100 200 300
```

Out[2]:
```
'\nfrom sys import argv\nsum = 0\nfor i in range(2, len(argv)):\n    sum
+= float(argv[i])\nprint(\n    "The average score for {0} is {1:.2f}"\n
.format(argv[1], sum/(len(argv)-2))\n)\n\n'
```

In [3]:
```
# Solución
# A
```

Question 2 (Data Aggregates)

What is the expected output of the following code?

```
data1 = '1', '2'
data2 = ('3', '4')
print(data1 + data2)
```

A. `('1', '2', '3', '4')`

B. `(1, 2, 3, 4)`

C. `['1', '2', '3', '4']`

D. The code is erroneous

Solution 2

In [5]:
```
# ejemplos previos
```

In [6]:
```
d1 = 1,2
d2 = (1, 2)
```

In [7]:
```
print('d1:', d1)
print('d2:', d2)
print('type(d1)', type(d1))
print('type(d2)', type(d2))
print('d1+d2:', d1+d2)
```

```
d1: (1, 2)
d2: (1, 2)
type(d1) <class 'tuple'>
type(d2) <class 'tuple'>
d1+d2: (1, 2, 1, 2)
```

In [8]:
```
data1 = '1', '2'
data2 = ('3', '4')
print(data1 + data2)
```

```
('1', '2', '3', '4')
```

In [9]:

```
# Solución
# A
```

Question 3 (Operators)

What is the expected output of the following code?

```
x, y, z = 3, 2, 1
z, y, x = x, y, z
print(x, y, z)
```

A. `2 1 3`

B. `3 2 1`

C. `1 2 2`

D. `1 2 3`

Solution 3

In [11]:

```
x, y, z = 3, 2, 1
# x = 3, y = 2, z = 1
z, y, x = x, y, z
# z = 3, y = 2, x = 1
print(x, y, z) # 1 2 3
```

```
1 2 3
```

In [12]:

```
# Solución
# D
```

Question 4 (Error Handling)

What is the expected output of the following code?

```
try:
    raise Exception
except BaseException:
    print('1')
except Exception:
    print('2')
except:
    print('3')
```

A. `3`

B. The code is erroneous

C. `1`

D. `2`

Solution 4

In [14]:

```
try:
    raise Exception
except BaseException:
    print('1')
except Exception:
    print('2')
except:
    print('3')
```

```
1
```

In [15]:

```
# Solución
# C
```

Question 5 (Operators)

What is the expected output of the following code?

```
nums = [3, 7, 23, 42]
alphas = ['p', 'p', 'm', 'j']

print(nums is alphas)
print(nums == alphas)

nums = alphas

print(nums is alphas)
print(nums == alphas)
```

A.

```
False
True
True
True
```

B.

```
False
True
False
True
```

C.

```
True
False
True
False
```

D.

```
False
False
True
True
```

Solution 5

In [17]:

```
nums = [3, 7, 23, 42]
alphas = ['p', 'p', 'm', 'j']

print(nums is alphas)  # False
print(nums == alphas)  # False

nums = alphas

# nums = ['p', 'p', 'm', 'j']
# alphas = ['p', 'p', 'm', 'j']

print(nums is alphas)   # True
print(nums == alphas)   # True
```

```
False
False
True
True
```

In [18]:

```
# Solución
# D
```

Question 6 (Control Flow)

You are coding a math utility by using Python. You are writing a function to compute roots. The function must meet the following requirements:

- If a is non-negative, `return a ** (1 / b)`
- If a is negative and even, `return 'Result is an imaginary number'`
- If a is negatitve and odd, `return -(-a) ** (1 / b)`

Which of the following functions meets the requirements?

A.

```
def safe_root(a,b):
    if a % 2 == 0:
        answer = a ** (1/b)
    elif a>=0:
        answer = 'Result is an imaginary number'
    else:
        answer = -(-a) ** (1/b)
    return answer
```

B.

```
def safe_root(a,b):
    if a >= 0:
        answer = a ** (1/b)
    elif a % 2 == 0:
        answer = 'Result is an imaginary number'
    else:
        answer = -(-a) ** (1/b)
    return answer
```

C.

```
def safe_root(a,b):
    if a >= 0:
        answer = -(-a) ** (1/b)
    elif a % 2 == 0:
        answer = 'Result is an imaginary number'
    else:
        answer = a ** (1/b)
    return answer
```

D.

```
def safe_root(a,b):
    if a % 2 == 0 :
        answer = -(-a) ** (1/b)
    elif a >= 0:
        answer = 'Result is an imaginary number'
    else:
        answer = a ** (1/b)
    return answer
```

Solution 6

In [20]:

```
# If a is non-negative, return a ** (1 / b)
# If a is negative and even, return 'Result is an imaginary number'
# If a is negatitve and odd, return -(-a) ** (1 / b)


def safe_root(a,b):
    if a >= 0:
        answer = a ** (1/b)
    elif a % 2 == 0:
        answer = 'Result is an imaginary number'
    else:
        answer = -(-a) ** (1/b)
    return answer
```

comprobaciones

In [21]:

```
# If a is non-negative, return a ** (1 / b)
# NOTA: NO PONER B=0
safe_root(2,1)
```

Out[21]:

```
2.0
```

In [22]:

```
# a ** (1 / b)
a=2
b=1
a ** (1 / b)
```

Out[22]:

```
2.0
```

In [23]:

```
# If a is negative and even, return 'Result is an imaginary number'
safe_root(-2,1)
```

Out[23]:

```
'Result is an imaginary number'
```

In [24]:

```
# If a is negatitve and odd, return -(-a) ** (1 / b)
safe_root(3,1)
```

Out[24]:

```
3.0
```

In [25]:

```
# -(-a) ** (1 / b)
a=3
b=1
-(-a) ** (1 / b)
```

Out[25]:

```
3.0
```

In [26]:

```
# Solución
# B
```

Question 7

Which of the following sentences is true?

```
str1 = 'Peter'
str2 = str1[:]
```

A. str2 is longer than str1

B. str1 is longer than str2

C. str1 and str2 are different (but equal) strings

D. str1 and str2 are different names of the same string

Solution 7 (Data Aggregates)

In [28]:

```
str1 = 'Peter'
str2 = str1[:]

print('str1:', str1)
print('str2:', str2)
print('str1 is str2:', str1 is str2)
print('str1 == str2:', str1 == str2)
```

```
str1: Peter
str2: Peter
str1 is str2: True
str1 == str2: True
```

In [29]:

```
# Solución
# D
```

Question 8 (Error Handling)

What is the expected behavior of the following program if the user enters 0?

```
value = input("Enter a value: ")
print(10/value)
```

A. The program will raise the ValueError exception

B. The program will raise the ZeroDivisionError exception

C. The program will raise the TypeError exception

D. The program will output 0 to the console

Solution 8

In [31]:

```
"""
value = input("Enter a value: ")
print(10/value)
"""

# TypeError: unsupported operand type(s) for /: 'int' and 'str'

# OJO, NO DICE INT(INPUT)
```

Out[31]:

```
'\nvalue = input("Enter a value: ")\nprint(10/value)\n'
```

In [32]:

```
# Solución
# C
```

Question 9 (Functions)

What is the expected output of the following code?

```
def fun():
    return 3

def add(n):
    return fun() + n

print(add(3))
```

A. `3`

B. `9`

C. `6`

D. The program will cause an error

Solution 9

In [34]:

```
def fun():
    return 3

def add(n):
    return fun() + n
            # 3   + 3
print(add(3)) # 6
```

```
6
```

In [35]:

```
# Solución
# C
```

Question 10 (Control Flow)

You are developing a Python application for an online product distribution company.

You need the program to iterate through a list of products and escape when a target product ID is found.

```
productIdList = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
index = 0

??? index < 10:
    print(productIdList[index])
    if productIdList[index] == 6:
        ???
    else:
       index += 1
```

What would you insert instead of ??? and ???

A.

```
while
for
```

B.

```
if
break
```

C.

```
for
break
```

D.

```
for
while
```

E.

```
while
break
```

F.

```
break
while
```

Solution 10

In [37]:

```
productIdList = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
index = 0

while index < 10:
    print(productIdList[index])
    if productIdList[index] == 6:
        break
    else:
        index += 1


# What would you insert instead of ??? and ???
```

```
0
1
2
3
4
5
6
```

In [38]:

```
# Solución
# E
```

Question 11 (Functions)

What is the expected output of the following code?

```
def func(data):
    data = [7, 23, 42]
    print('Function scope: ', data)

data = ['Peter', 'Paul', 'Mary']
func(data)
print('Outer scope: ', data)
```

A.

```
Function scope: [7, 23, 42]
Outer scope: ['Peter', 'Paul', 'Mary']
```

B.

```
None of the above
```

C.

```
Function scope: [7, 23, 42]
Outer scope: [7, 23, 42]
```

D.

```
Function scope: ['Peter', 'Paul', 'Mary']
Outer scope: ['Peter', 'Paul', 'Mary']
```

Solution 11

In [40]:
```
def func(data):
    data = [7, 23, 42]
    print('Function scope: ', data)
    # function scope: [7, 23, 42]


data = ['Peter', 'Paul', 'Mary']
func(data)
print('Outer scope: ', data)
# outer scope: ['Peter', 'Paul', 'Mary']

# data en la función no es global


# quedará:
# function scope: [7, 23, 42]
# outer scope: ['Peter', 'Paul', 'Mary']
```

```
Function scope:  [7, 23, 42]
Outer scope:  ['Peter', 'Paul', 'Mary']
```

In [41]:
```
# Solución
# A
```

Question 12 (Functions)

What is the expected output of the following code?

```
def func(x=2, y=3):
    return x * y

print(func(y=2))
```

A. `6`

B. The code is erroneous

C. `4`

D. `2`

Solution 12

In [43]:
```
def func(x=2, y=3):
    return x * y

print(func(y=2))


# 2 * 2 = 4
```

```
4
```

In [44]:

```
# Solución
# C
```

Question 13 (Operators)

What is the data type of x, y, z after executing the following snippet?

```
x = 23 + 42
y = '23' + '42'
z = '23' * 7
```

A.

```
x is int,
y and z are invalid declarations
```

B.

```
int, str, int
```

C.

```
int, int, int
```

D.

```
int, str, str
```

Solution 13

In [46]:

```
x = 23 + 42
y = '23' + '42'
z = '23' * 7

print("x: ", x)              # 65
print("y: ", y)              # '2342'
print("z: ", z)              # 23 (7 veces), '23232323232323'
print('type(x):', type(x))  # int
print('type(y):', type(y))  # str
print('type(z):', type(z))  # str
```

```
x:  65
y:  2342
z:  23232323232323
type(x): <class 'int'>
type(y): <class 'str'>
type(z): <class 'str'>
```

In [47]:

```
# Solución
# D
```

Question 14 (Operators)

What is the expected output of the following code?

```
print(3 * 'abc' + 'xyz')
```

A. `3abcxyz`

B. `abcxyzabcxyzabcxyz`

C. `abcabcabcxyz`

D. `abcabcxyzxyz`

Solution 14

In [49]:

```
# 3 veces abc y después xyz

print(3 * 'abc' + 'xyz')

# abcabcabcxyz
```

```
abcabcabcxyz
```

In [50]:

```
# Solución
# C
```

Question 15 (Data Types)

How many arguments can the `print()` function take?

A. Any number of arguments (excluding zero)

B. Any number of arguments (including zero)

C. Not more than seven arguments

D. Just one argument

Solution 15

In [52]:

```
print()

# print es posible
```

In [53]:

```
# Solución
# B
```

Question 16 (Basics)

The `print()` function can output values of:

A. not ore than five arguments

B. any number of arguments (including zero)

C. just one argument

D. any number of arguments (excluding zero)

Solution 16

In [55]:

```
print()
```

In [56]:

```
# Solución
# B
```

Question 17 (Operators)

What would you insert instead of ??? so that the program prints True to the monitor?

```
x = 'Peter'
y = 'Peter'
res = ???
print(res)
```

A. `x != y`

B. `x is y`

C. `x is not y`

D. `x < y`

Solution 17

In [58]:

```
x = 'Peter'
y = 'Peter'
res = x is y
print(res)
```

```
True
```

In [59]:

```
# Solución
# B
```

Question 18 (Functions)

The following snippet:

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

print(func(b=2, 2))
```

A. will output `None`

B. will output `2`

C. will output `4`

D. is erroneous

Solution 18

In [61]:

```
"""
def func(a, b):
    return b ** a

print(func(b=2, 2))
"""

# SyntaxError: positional argument follows keyword argument
```

Out[61]:

```
'\ndef func(a, b):\n    return b ** a\n \nprint(func(b=2, 2))  \n'
```

In [62]:

```
# Solución
# D
```

Question 19 (Data Types)

Which of the following statements can be used to return the length of the given string str?

A. `str._len_()`

B. `len(str)`

C. `size(str)`

D. `str.size()`

Solution 19

In [64]:

```
string = 'casa'
len(string)

# realmente ningún string debería llamarse str
# pero sería la opción b
```

Out[64]: 4

In [65]:

```
# Solución
# B
```

Question 20 (Control Flow)

The following is a program to validate customer numbers.

```
customer_number = input('Enter the employee number (dd-
ddd-dddd): ')
parts = customer_number.split('-')
valid = False
if len(parts) == 3:
  if len(parts[0]) == 2 and len(parts[1]) == 3 and
len(parts[2]) == 4:
    if parts[0].isdigit() and parts[1].isdigit() and
parts[2].isdigit():
      valid = True
print(valid)
```

The number may only contain numbers and dashes. The number must have the right format (dd-ddd-dddd). What is true about this programm?

A. The program works properly

B. There will be a `SyntaxError`

C. There will be an `AttributeError`

D. There will be no error but there will be an unwanted result

Solution 20

In [67]:

```
customer_number = input('Enter the employee number (dd-ddd-dddd): ')
parts = customer_number.split('-')
valid = False
if len(parts) == 3:
    if len(parts[0]) == 2 and len(parts[1]) == 3 and len(parts[2]) == 4:
        if parts[0].isdigit() and parts[1].isdigit() and parts[2].isdigit
            valid = True
print(valid)
```

```
Enter the employee number (dd-ddd-dddd): 10-101-1010
True
```

In [68]:

```
# Solución
# A
```

Question 21 (Data Types)

You want to write a program that asks the user for a value.

For the rest of the program you need a whole number, even if the user enters a decimal value.

What would you have to write?

A. `num = int('How many do you need?')`

B. `num = float(input('How many do you need?'))`

C. `num = str(input('How many do you need?'))`

D. `num = int(float(input('How many do you need?')))`

Solution 21

In [70]:

```
# D
num = int(float(input('How many do you need?')))
num
```

```
How many do you need?10
```

Out[70]:

```
10
```

In [71]:

```
# Solución
# D
```

Question 22 (Data Aggregates)

Which one of the lines should you put in the snippet below to match the expected output?

Expected output:

```
[4, 1, 7, 2, 'A']
```

Code:

```
list = ['A', 2, 7, 1, 4]
enter code here
print(list)
```

A. `reversed(list)`

B. `reverse(list)`

C. `list.reverse()`

D. `list.reversed()`

Solution 22

In [73]:

```
list = ['A', 2, 7, 1, 4]
# enter code here
list.reverse()

print(list)

# Expected output:
# [4, 1, 7, 2, 'A']
```

```
[4, 1, 7, 2, 'A']
```

In [74]:

```
# Solución
# C
```

Question 23 (Basics)

What is the expected output of the following code?

```
x = '\\\'
print(len(x))
```

A. The code is erroneous

B. `2`

C. `1`

D. `3`

Solution 23

In [76]:

```
"""
x = '\\\'

print(len(x))
"""

# SyntaxError: EOL while scanning string literal
```

Out[76]:

```
"\nx = '\\\'\n\nprint(len(x))\n"
```

In [77]:

```
# Solución
# A
```

Question 24 (Control Flow)

Consider the following code.

```
data = ['Peter', 'Paul', 'Mary', 'Jane']
res = 0
```

Which of the following code snippets will expand the code, so that 100 will be printed to the monitor?

Choose two

A.

```
for i in ('Peter', 'Steve', 'Jane'):
    if i not in data:
        res += 50
print(res)
```

B.

```
for i in ('Peter', 'Steve', 'Jane'):
    if i in data:
        res += 50
print(res)
```

C.

````
for i in ('Peter', 'Steve', 'Jane'):
    if i not in data:
        res += 100
print(res)```
````

D.

````
for i in ('Peter', 'Steve', 'Jane'):
    if i in data:
        res += 100
print(res)
````

Solution 24

In [79]:

````python
# A.

data = ['Peter', 'Paul', 'Mary', 'Jane']
res = 0

for i in ('Peter', 'Steve', 'Jane'):
    if i not in data:
        res += 50
print(res)
````

````
50
````

In [80]:

````python
# B

data = ['Peter', 'Paul', 'Mary', 'Jane']
res = 0

for i in ('Peter', 'Steve', 'Jane'):
    if i in data:
        res += 50
print(res)
````

````
100
````

In [81]:

````python
# C

data = ['Peter', 'Paul', 'Mary', 'Jane']
res = 0

for i in ('Peter', 'Steve', 'Jane'):
    if i not in data:
        res += 100
print(res)
````

````
100
````

In [82]:

````python
# D

data = ['Peter', 'Paul', 'Mary', 'Jane']
res = 0

for i in ('Peter', 'Steve', 'Jane'):
    if i in data:
````

```
        res += 100
print(res)
```

```
200
```

In [83]:

```
# Solución
# B y C
```

Question 25 (Data Types)

Consider the following Python code:

```
name = 'Peter'
age = 23
flag = True
```

What are the types of the variables name, age and flag?

A. `int, bool, char`

B. `str, int, int`

C. `str, int, bool`

D. `float, bool, str`

Solution 25

In [85]:

```
name = 'Peter'
age = 23
flag = True

print('type(name):', type(name))
print('type(age):', type(age))
print('type(flag):', type(flag))
```

```
type(name): <class 'str'>
type(age): <class 'int'>
type(flag): <class 'bool'>
```

In [86]:

```
# Solución
# C
```

Question 26 (Control Flow)

How many stars will the following snippet print to the monitor?

```
for i in range(1):
    print('*')
else:
    print('*')
```

A. one

B. zero

C. two

D. three

Solution 26

In [88]:

```
for i in range(1): # 0
    print('1')
    print('*')
    print('salto a else')
else:
    print('2')
    print('*')
```

```
1
*
salto a else
2
*
```

In [89]:

```
for i in range(1):
    print('*')
else:
    print('*')
```

```
*
*
```

In [90]:

```
# Solución
# C
```

Question 27 (Data Aggregates)

What is the expected output of the following code?

```
data = {'Peter': 30, 'Paul': 31}
print(list(data.keys()))
```

A. `('Peter': 30, 'Paul': 31)`

B. `['Peter', 'Paul']`

C. `('Peter', 'Paul')`

D. `['Peter': 30, 'Paul': 31]`

Solution 27

In [92]:

```
# Cosas previas
```

In [93]:

```
data = {'Peter': 30, 'Paul': 31}
data.keys()
```

Out[93]:

```
dict_keys(['Peter', 'Paul'])
```

In [94]:

```
# El propio ejercicio
```

In [95]:

```
"""data = {'Peter': 30, 'Paul': 31}
print(list(data.keys()))"""

# me da error cuando ejecuto todo de golpe
```

Out[95]:

```
"data = {'Peter': 30, 'Paul': 31}\nprint(list(data.keys()))"
```

In [96]:

```
# Solución
# B
```

Question 28 (Functions)

What is the expected output of the following code?

```
def func(n):
    s = ''
    for i in range(n):
        s += '*'
        yield s

for x in func(3):
    print(x, end='')
```

A. The code is erroneous

B. `*`

C. `***`

D. `******`

Solution 28

In [98]:

```
def func(n):
    s = ''
    for i in range(n):     # range(3) => 0,1,2
        s += '*'           # s = s + *
        yield s


for x in func(3):
    print(x, end='')
```

```
******
```

In [99]:

```
# Solución
# D
```

Question 29 (Control Flow)

How many stars will the following snippet send to the console?

```
for i in range(-1, 1):
    print("*")
```

A. two

B. three

C. one

Solution 29

In [101…

```
for i in range(-1, 1):   # -1, 0
    print("*")           # 2 *
```

```
*
*
```

In [102…

```
# Solución
# A
```

Question 30 (Data Types)

What is the expected output of the following code if the user enters 2 and 4?

```
x = int(input())
y = int(input())
print(x + y)
```

A. `4`

B. `6`

C. `24`

D. `2`

Solution 30

In [104…

```
# 2, 4
x = int(input())
y = int(input())
print(x + y)          # 6
```

```
2
4
6
```

In [105…

```
# Solución
# B
```

Gracias por la atención

Isabel Maniega