Contenido creado por: Isabel Maniega

Question 1 (Control Flow)

Consider the following code.

```
for i in range(5, 0, ???):
    print(i, i, i, i, i)
```

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

```
1. 5 5 5 5 5
2. 4 4 4 4 4
3. 3 3 3 3 3
4. 2 2 2 2 2
5. 1 1 1 1 1
```

A. `1`

B. `0`

C. `None`

D. `-1`

Solution 1

In [2]:

```
for i in range(5, 0, -1):
    print(i, i, i, i, i)
```

```
5 5 5 5 5
4 4 4 4 4
3 3 3 3 3
2 2 2 2 2
1 1 1 1 1
```

In [3]:

```
# Solución
# D
```

Question 2 (Basics)

A value returned by the `input()` function is:

A. an integer

B. a float

C. a string

Solution 2

In [5]:

```
x = input("introduce un texto: ")

print('texto:', x, 'y type(x):', type(x))
```

```
introduce un texto: hola mundo
texto: hola mundo y type(x): <class 'str'>
```

In [6]:

```
# Solución
# C
```

Question 3 (Control Flow)

What is the expected output of the following code?

```
x = (1, 4, 7, 9, 10, 11)
y = {2: 'A', 4: 'B', 6: 'C', 8: 'D', 10: 'E', 12: 'F'}
res = 1

for z in x:
    if z in y:
        continue
    else:
        res += z

print(res)
```

A. `30`

B. `31`

C. `29`

D. `28`

Solution 3

In [8]:

```
x = (1, 4, 7, 9, 10, 11)

y = {2: 'A',
     4: 'B',
     6: 'C',
     8: 'D',
     10: 'E',
     12: 'F'}

res = 1   # OJO!!, NO EMPIEZA EN 0

for z in x:
    if z in y:
        continue
    else:
        res += z   # res = res + z
        # 1, 7, 9, 11
        # (1) + 1 + 7 + 9 + 11 = 29
        # EMPIEZA EN 1 + TODOS ESOS ELEMENTOS

print(res)  # 5 ELEMENTOS => 29
```

```
29
```

In [9]:

```
# Solución
# C
```

Question 4 (Data Aggregates)

Which of the following sentences are true about the code?

```
nums = [1, 2, 3]
vals = nums
```

(Select two answers)

A. vals is longer than nums

B. nums and vals are different lists

C. nums has the same length than vals

D. nums and vals are different names of the same list

Solution 4

In [11]:

```
nums = [1, 2, 3]
vals = nums

print('nums:', nums, 'len(nums): ', len(nums))
print('vals:', vals, 'len(vals): ', len(vals))
```

```
nums: [1, 2, 3] len(nums):  3
vals: [1, 2, 3] len(vals):  3
```

In [12]:

```
# Solución
# C y D
```

Question 5 (Data Aggregates)

What is the expected output of the following code?

```
data = ['Peter', 'Paul', 'Mary']
print(data[int(-1 / 2)])
```

A. `Paul`

B. `Mary`

C. The code is erroneous

D. None of the above

E. `Peter`

Solution 5

In [14]:

```
int(0.5), int(-0.5), int(3/2), int(-3/2)
```

Out[14]:

```
(0, 0, 1, -1)
```

In [15]:

```
data = ['Peter', 'Paul', 'Mary']

print(data[int(-1 / 2)])

# data[0] => Peter
```

```
Peter
```

In [16]:

```
# Solución
# E
```

Question 6 (Functions)

What is the expected output of the following code?

```
def func(x, y, z):
    return x + 4 * y + 5 * z

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

A. `24`

B. The code is erroneous

C. `23`

D. `80`

Solution 6

In [18]:

```python
def func(x, y, z):
    return x + 4 * y + 5 * z

print(func(1, z=2, y=3))

# func(1, 3, 2)
# 1 + 4*3 + 5*2 = 1 + 12 + 10 = 23
```

```
23
```

In [19]:

```python
# Solución
# C
```

Question 7 (Operators)

What is the expected output of the following code?

```python
res = str(bool(1) + float(12) / float(2))
print(res)
```

A. `6`

B. `7`

C. The code is erroneous

D. `7.0`

E. `6.0`

Solution 7

In [21]:

```python
# conceptos previos
```

In [22]:

```python
bool(1), bool(10), bool(0), bool(-10)
```

Out[22]:

```
(True, True, False, True)
```

In [23]:

```python
for i in range(-5,6,1):
    print('i:', i, 'bool(i):', bool(i))
```

```
i: -5 bool(i): True
i: -4 bool(i): True
i: -3 bool(i): True
i: -2 bool(i): True
i: -1 bool(i): True
i: 0 bool(i): False
i: 1 bool(i): True
i: 2 bool(i): True
i: 3 bool(i): True
i: 4 bool(i): True
i: 5 bool(i): True
```

In [24]:

```
float(12), float(2.5), float(2)
```

Out[24]:

```
(12.0, 2.5, 2.0)
```

In [25]:

```
# el propio ejercicio
```

In [26]:

```
res = str(bool(1) + float(12) / float(2))
print(res)
```

```
7.0
```

In [27]:

```
# res = str(bool(1) + (float(12) / float(2)))
# primero "* y /", "después + y -"
# print(res)
```

In [28]:

```
# Solución
# D
```

Question 8 (Data Types)

The ABC company is building a basketball court for its employees to improve company morale.

You are creating a Python program that employees can use to keep track of their average score.

The program must allow users to enter their name and current scores.

The program will output the user name and the user's average score. The output must meet the following requirements:

The user name must be left-aligned

If the user name has fewer than 20 characters, additional space must be added to the right

The average score must have three places to the left of the decimal point and one place to the right of the decimal (xxx.x)

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

```
name = input('What is your name?')
sum = 0
score = 0
count = 0
while score != -1:
    score = int(input('Enter your scores: (-1 to end)'))
    if score == -1:
        break
    sum += score
    count += 1
average = sum / count
print('???, your average score is: ???' % (name, average))
```

A.

```
%-20s
%1.4f
```

B.

```
-20d
4.1f
```

C.

```
-20i
4.1
```

D.

```
-20s
4.1f
```

E.

```
-20f
%1.4s
```

F.

```
-20f
%4,1s
```

Solution 8

In [30]:

```
name = input('What is your name?')
sum = 0
score = 0
count = 0
while score != -1:
    score = int(input('Enter your scores: (-1 to end)'))
    if score == -1:
        break
    sum += score
    count += 1
average = sum / count
print('%-20s, your average score is: %4.1f' % (name, average))
```

```
What is your name?Jose
Enter your scores: (-1 to end)1
Enter your scores: (-1 to end)3
Enter your scores: (-1 to end)5
Enter your scores: (-1 to end)-1
Jose                , your average score is:  3.0
```

In [31]:

```
# Solución
# D
```

Question 9 (Control Flow)

What is the expected output of the following snippet?

```
s = 'python'
for i in range(len(s)):
    i = s[i].upper()
print(s, end='')
```

A. `P y t h o n`

B. `P Y T H O N`

C. `python`

D. The code is erroneous

E. `PYTHON`

F. `Python`

Solution 9

In [33]:

```
s = 'python'
for i in range(len(s)):   # len(s)=6 # range(6)-> 0,1,2,3,4,5
    i = s[i].upper()      # PYTHON (mayúsculas)
print(s, end='')          # PYTHON (sin espacios en la misma línea)
# PERO CAMBIOS NO AÑADIDOS AL STRING

# Solución: python
```

```
python
```

In [34]:

```
# Solución
# C
```

Question 10 (Basics)

You execute the following command in the terminal.

```
python index.py Hello
```

You want the command to print out `Hello`

What has to be inside of index.py?

A.

```
from sys import args
print(args[0])
```

B.

```
from sys import argv
print(argv[0])
```

C.

```
from sys import args
print(args[1])
```

D.

```
from sys import argv
print(argv[1])
```

Solution 10

In [36]:

```
# Solución
# D
```

Question 11 (Data Aggregates)

Which function does in-place reversal of objects in a list?

A. `list.reverse()`

B. `list.sort([func])`

C. `list.remove(obj)`

D. `list.pop(obj=list[-1])`

Solution 11

In [38]:

```
listado = [1,2,3]
listado.reverse()
print(listado)
```

```
[3, 2, 1]
```

In [39]:

```
# Solución
# A
```

Question 12 (Operators)

Which expression evaluates to `7` ?

A. `9 % 3 + 7`

B. `9 - 3 * 7`

C. `9 / 3 * 7`

D. `9 // 3 - 7`

Solution 12

In [41]:

```
# A. 9 % 3 + 7   ==>  0 + 7 = 7
print(9 % 3 + 7)
# B. 9 - 3 * 7   ==>  9 - 21 = -12
print(9 - 3 * 7)
# C. 9 / 3 * 7   ==>  3 /* 7 =  21 (IZQUIERDA A DERECHA) !!!!!
print(9 / 3 * 7)
# D. 9 // 3 - 7  ==>  3 - 7 = -4
print(9 // 3 - 7)
```

```
7
-12
21.0
-4
```

In [42]:

```
# Solución
# A
```

Question 13 (Functions)

What is the expected output of the following code?

```
def fun(n):
    x = []
    for i in range(n):
        x.append(i)
    return x

print(fun(4))
```

A.

```
0
1
2
3
```

B. `[4, 4, 4, 4]`

C. The program will cause an error

D. `[0, 1, 2, 3]`

Solution 13

In [44]:

```
def fun(n):
    x = []                  # x = []
    for i in range(n):      # range(4) => 0, 1, 2, 3
        x.append(i)         # [0,1,2,3] les va apendizando
    return x

print(fun(4))
```

```
[0, 1, 2, 3]
```

In [45]:

```
# Solución
# D
```

Question 14 (Data Types)

What is the expected output of the following code if the user enters `4` and `7`?

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

A. `7777`

B. `47`

C. `4444444` # 4 (7 veces)

D. `28`

Solution 14

In [47]:

```
# 4 y 7

x = input()
y = int(input())
print(x * y)      # el 4 (7 veces)
```

```
4
7
4444444
```

In [48]:

```
# Solución
# C
```

Question 15 (Data Aggregates)

The second assignment:

```
vals = [0, 1, 2]
vals[0], vals[1] = vals[1], vals[2]
```

A. doesn't change the list's length

B. shortens the list

C. extends the list

Solution 15

In [50]:

```
vals = [0, 1, 2]
print('longitud inicial:', len(vals))

vals[0], vals[1] = vals[1], vals[2]

# vals[0] = vals[1] => vals[0] = 1  ==>  [1, 1, 2]
# vals[1] = vals[2] => vals[1] = 2  ==>  [1, 2, 2]

print(vals)
print('longitud final:', len(vals))
```

```
longitud inicial: 3
[1, 2, 2]
longitud final: 3
```

In [51]:

```
# Solución
# A
```

Question 16 (Basics)

What is the expected behavior of the following program?

```
prin("Goodbye!")
```

A. The program will generate an error message on the screen

B. The program will output `("Goodbye!")`

C. The program will output `Goodbye!`

D. The program will output `"Goodbye!"`

Solution 16

In [53]:

```
# prin no va a ejecutar
"""
prin("Goodbye!")
"""

# NameError: name 'prin' is not defined
```

Out[53]:

```
'\nprin("Goodbye!")\n'
```

In [54]:

```
# Solución
# A
```

Question 17 (Control Flow)

Consider the following code.

```
x = 42
y = 7
data = "I'm gonna make him an offer he can't refuse."
```

Which of the following expressions will evaluate to `19`?

A. `7 if len(data) > 19 else 6`

B. `data.rfind('an') if data else None`

C. `19 if None else x / y`

D. `data.find('an') if data else None`

Solution 17

In [56]:

```
x = 42
y = 7
data = "I'm gonna make him an offer he can't refuse."
# A
7 if len(data) > 19 else 6
```

Out[56]:

```
7
```

In [57]:

```
x = 42
y = 7
data = "I'm gonna make him an offer he can't refuse."
# B
data.rfind('an') if data else None
```

Out[57]:

```
32
```

In [58]:

```
x = 42
y = 7
data = "I'm gonna make him an offer he can't refuse."
# C
19 if None else x / y
```

Out[58]:

```
6.0
```

In [59]:

```
x = 42
y = 7
data = "I'm gonna make him an offer he can't refuse."
# D
data.find('an') if data else None
```

Out[59]:

```
19
```

In [60]:

```
# Solución
# D
```

Question 18 (Functions)

What is the expected output of the following code?

```
def func(x):
    if x == 0:
        return 0
    return x + func(x - 1)

print(func(3))
```

A. `3`

B. The code is erroneous

C. `6`

D. `1`

Solution 18

In [62]:

```
def func(x):
    if x == 0:
        return 0
    return x + func(x - 1)
      #    3 + func(3 - 1)
      #    3 + func(2)
      #    3 + 2 + f(1)
      #    3 + 2 + 1 + f(0)
      #    3 + 2 + 1 + 0 (return 0 cuando ya es 0 x==0)

print(func(3))  # 6
```

```
6
```

In [63]:

```
# Solución
# C
```

Question 19 (Control Flow)

You work for a company that distributes media for all ages.

You are writing a function that assigns a rating based on a user's age. The function must meet the following requirements:

- Anyone 18 years old or older receives a rating of A
- Anyone 13 or older, but younger than 18 receives a rating of T
- Anyone 12 years old or younger receives a rating of C

If the age is unknown, the rating is set to C

```
def get_rating(age):
    rating = ''
    if              # Line-3
    elif            # Line-4
    elif            # Line-5
    else            # Line-6
    return rating
```

Which of the following should you insert on Line-3 to Line-6?

A.
```

age < 13: rating = 'C': # Line-3

age == None: rating = 'C' # Line-4

age < 18: rating = 'T' # Line-5

: rating = 'A' # Line-6
```

B.
```

age < 18: rating = 'T': # Line-3

age < 13: rating = 'C' # Line-4

age == None: rating = 'C' # Line-5

: rating = 'A' # Line-6
```

C.
```

rating = 'A': # Line-3 age < 18: rating = 'T' # Line-4 age < 13: rating = 'C' # Line-5

age == None: rating = 'C' # Line-6
```

D.

```
age == None: rating = 'C':  # Line-3
age < 18: rating = 'T'      # Line-4
age < 13: rating = 'C'      # Line-5
 : rating = 'A'             # Line-6
```

E.

```
age == None: rating = 'C':  # Line-3
age < 13: rating = 'C'      # Line-4
age < 18: rating = 'T'      # Line-5
 : rating = 'A'             # Line-6
```

F.

```
 : rating = 'A'             # Line-3
age < 13: rating = 'C'      # Line-4
age < 18: rating = 'T'      # Line-5
age == None: rating = 'C'   # Line-6
```

Solution 19

B

In [65]:
```
# Anyone 18 years old or older receives a rating of A
# Anyone 13 or older, but younger than 18 receives a rating of T
# Anyone 12 years old or younger receives a rating of C
# If the age is unknown, the rating is set to C


def get_rating(age):
    rating = ''
    if age < 18:
        rating = 'T'
    elif age < 13:
        rating = 'C'
    elif age == None:
        rating = 'C'
    else:
        rating = 'A'

    return rating
```

In [66]:

```
print('menores de 12:', get_rating(5))
print('13 a 18:', get_rating(14))
print('mayores de 18:', get_rating(20))


# NO CUMPLE
```

```
menores de 12: T
13 a 18: T
mayores de 18: A
```

E.

In [67]:

```
# age == None: rating = 'C':  # Line-3
# age < 13: rating = 'C'      # Line-4
# age < 18: rating = 'T'      # Line-5
#   : rating = 'A'            # Line-6
```

In [68]:

```
def get_rating(age):
    rating = ''
    if age == 'None':
        rating = 'C'
    elif age < 13:
        rating = 'C'
    elif age < 18:
        rating = 'T'
    else:
        rating = 'A'

    return rating
```

In [69]:

```
print('menores de 12:', get_rating(5))
print('13 a 18:', get_rating(14))
print('mayores de 18:', get_rating(20))

# Anyone 18 years old or older receives a rating of A
# Anyone 13 or older, but younger than 18 receives a rating of T
# Anyone 12 years old or younger receives a rating of C
# If the age is unknown, the rating is set to C
```

```
menores de 12: C
13 a 18: T
mayores de 18: A
```

In [70]:

```
# SI cumple
```

In [71]:

```
# Solución
# E
```

Question 20 (Functions)

Consider the following code.

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

With sentence describes the function the best?

A function defined like `function()` ...

A. may be called with any number of arguments (including zero)

B. must be called without arguments

C. may be called without any argument, or just with one

D. must be called with exactly one argument

Solution 20

In [73]:

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

# llamada a la función
function()
```

Out[73]:

```
0
```

In [74]:

```
# Solución
# C
```

Question 21 (Basics)

Insert the correct piece of code so that the program produces the expected output.

Expected output:

```
Andy
Brown
```

Code:

```
insert code here
```

A. `print("Andy Brown", sep=" ")`

B. `print("Andy\nBrown")`

C. `print("Andy Brown", end=" ")`

D. `print(Andy\nBrown)`

Solution 21

In [76]:

```
# B

print("Andy\nBrown")
```

```
Andy
Brown
```

In [77]:

```
# Solución
# B
```

Question 22 (Basics)

The `\n` digraph forces the `print()` function to:

A. break the output line

B. output exactly two characters: `\` and `n`

C. duplicate the character next to the digraph

D. stop its execution

Solution 22

In [79]:

```
print("hola \n mundo")
```

```
hola 
 mundo
```

In [80]:

```
# Solución
# A
```

Question 23 (Basics)

The compiled Python bytecode is stored in files having names ending with:

A. pyc

B. py

C. pyb

D. pc

Solution 23

```
In [82]: # pyc
```

```
In [83]: # Solución
         # A
```

Question 24 (Operators)

What is the expected output of the following code?

```
print('Mike' > 'Mikey')
```

A. `False`

B. `0`

C. `True`

D. `1`

Solution 24

```
In [85]: ord(' '), ord('y'), ord(' ') > ord('y')

Out[85]: (32, 121, False)
```

```
In [86]: print('Mike' > 'Mikey')

False
```

In [87]:

```
# Solución
# A
```

Question 25 (Control Flow)

You are designing a decision structure to convert a student's numeric grade to a letter grade. The program must assign a letter grade as specified as followed:

```
1: 90 through 100 -> A
2: 80 through  89 -> B
3: 70 through  79 -> C
4: 65 through  69 -> D
5:  0 through  64 -> F
```

- For example, if the user enters a 90 the output should be Your letter grade is A
- Likewise, if a user enters an 89 the output should be Your letter grade is B

```
# Letter Grade Converter
grade = int(input('Enter a numeric grade:'))
# Line-3
    letter_grade = 'A'
# Line-5
    letter_grade = 'B'
# Line-7
    letter_grade = 'C'
# Line-9
    letter_grade = 'D'
else:
    letter_grade = 'F'
print('Your letter grade is:', letter_grade)
```

Which of the following should you insert on

Line-3, Line-5, Line-7 and Line-9?

A.

```
if grade <= 90:    # Line-3
elif grade <= 80:  # Line-5
elif grade <= 70:  # Line-7
elif grade <= 65:  # Line-9
```

B.

```
if grade >= 90:    # Line-3
elif grade >= 80:  # Line-5
elif grade >= 70:  # Line-7
elif grade >= 65:  # Line-9
```

C.

```
if grade < 90:     # Line-3
elif grade < 80:   # Line-5
elif grade < 70:   # Line-7
elif grade < 65:   # Line-9
```

D.

```
if grade > 90:     # Line-3
elif grade > 80:   # Line-5
elif grade > 70:   # Line-7
elif grade > 65:   # Line-9
```

Solution 25

In [89]:

```
# B.
```

In [90]:

```
# 1: 90 through 100 -> A
# 2: 80 through  89 -> B
# 3: 70 through  79 -> C
# 4: 65 through  69 -> D
# 5:  0 through  64 -> F


# Letter Grade Converter
grade = int(input('Enter a numeric grade:'))
if grade >= 90:    # Line-3
    letter_grade = 'A'
elif grade >= 80:  # Line-5
    letter_grade = 'B'
elif grade >= 70:  # Line-7
    letter_grade = 'C'
elif grade >= 65:  # Line-9
    letter_grade = 'D'
else:
    letter_grade = 'F'
print('Your letter grade is:', letter_grade)
```

```
Enter a numeric grade:85
Your letter grade is: B
```

In [91]:

```
# Solución
# B
```

Question 26 (Operators)

Which of the following is the correct order of operator precedence?

A.

```
Exponents
Unary positive, negative, not Parentheses
Multiplication and Division
Addition and Subtraction
And
```

B.

```
Exponents
Unary positive, negative, not
Multiplication and Division
Addition and Subtraction
And
Parentheses
```

C.

```
Parentheses
Exponents
Unary positive, negative, not
Addition and Subtraction
Multiplication and Division
And
```

D.

```
Parentheses
Exponents
Unary positive, negative, not
Multiplication and Division
Addition and Subtraction
And
```

Solution 26

In [93]:

```
# Solución
# D
```

Question 27 (Error Handling)

What is the expected output of the following code?

```
num = '7' * '7'
print(num)
```

A. `49`

B. The code is erroneous

C. `77`

D. `7777777` (el numero 7 - 7 veces)

Solution 27

In [95]:
```
# ejemplos previos
```

In [96]:
```
num1 = '7' * 7
num2 = '7' + '7'
print(num1)
print(num2)
```

```
7777777
77
```

In [97]:
```
# el propio ejemplo
```

In [98]:
```
"""
num = '7' * '7'
print(num)
"""


# TypeError: can't multiply sequence by non-int of type 'str'
```

Out[98]:
```
"\nnum = '7' * '7'\nprint(num)\n"
```

In [99]:
```
# Solución
# B
```

Question 28 (Functions)

What is the output of the following snippet?

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

print(fun(out=2))
```

A. the snippet is erroneous

B. `2`

C. `4`

D. `6`

Solution 28

In [101…
```
def fun(inp=2, out=3):
    return inp * out    # 2 * 2

print(fun(out=2)) # 4
```
```
4
```

In [102…
```
# Solución
# C
```

Question 29 (Data Aggregates)

What is the output of the following snippet?

```
data = ['abc', 'def', 'abcde', 'efg']
print(max(data))
```

A. None of the above

B. `efg`

C. The code is erroneous

D. `def`

Solution 29

In [104…
```
data = ['abc', 'def', 'abcde', 'efg']
print(max(data))
```
```
efg
```

In [1]:
```
ord('a'), ord('d'), ord('e') # ==> 'efg' sera el mayor
```
Out[1]:
```
(97, 100, 101)
```

In [106…
```
# Solución
# B
```

Question 30 (Data Types)

Consider the following code.

```
data = eval(input('Input: '))
print('Output:', data)
```

Which of the inputs below would produce the specified output?

A.

```
Input = [x**2 for x in range(1,4)]
Output = [1, 4, 9]
```

B.

```
Input: Hello Python
Output: Hello Python
```

C. None of the above

D. Both are correct

Solution 30

In [108…

```
data = eval(input('Input: '))   # [x**2 for x in range(1,4)]
print('Output:', data)
```

```
Input: [x**2 for x in range(1,4)]
Output: [1, 4, 9]
```

In [109…

```
# Solución
# A
```

Gracias por la atención

Isabel Maniega