

1. Determina el valor de  $x$  al final del siguiente código:

1 / 1 punto

```
1  x = 1
2  i = 0
3  while i < 4:
4      x = x * 2
5      i += 1 # es lo mismo que i = i + 1
6  print(x)
```

- ☐ 1
- ☒ 16
- ☐ 4
- ☐ 8

☒ Correcto

Correcto. La variable  $x$  se va multiplicando por 2, lo que se repite 4 veces, dado que  $i$  comienza siendo 0, luego es 1, 2, y 3 (eso representa, entonces, 4 ciclos del while). Cuando  $i$  llega a ser 4, la condición del while ya no se cumple, y  $x$  se ha multiplicado por 2 cuatro veces (es decir, tomó los valores 2, 4, 8 y finalmente 16, que es el valor que tiene al final del programa).

2. Al final de este código:

1 / 1 punto

```
1  x = 48
2  y = 8
3  n = 0
4  while x > 0:
5      x = x - y
6      n = n + 1
```

El valor de  $n$  es:

- ☐  $x - y$
- ☒  $x // y$
- ☐  $x ** y$
- ☐  $y / x$

☒ Correcto

Correcto. En cada iteración del ciclo while, se va restando el valor  $y$  del valor de  $x$ . Es decir, en la primera iteración,  $x$  vale 40 ( $n=1$ ), en la segunda,  $x$  es 32 ( $n=2$ ), luego  $x$  es 24 ( $n=3$ ), 16 ( $n=4$ ), 8 ( $n=5$ ) y finalmente 0 ( $n=6$ ).

3. Considerando las variables del siguiente código:

1 / 1 punto

```
1  a = 5
2  b = 8
3  r = 0
4  while a > 0:
5      r = r + b
6      a = a - 1
7  print(r)
```

El *output* del mismo es equivalente a imprimir:

- ☒ a \* b
- ☐ b \*\* a
- ☐ a \*\* b
- ☐ a / b

☒ Correcto

Correcto. En cada iteración del ciclo while, se suma el valor de la variable *b* a la variable *r*. Esto se repite *a* veces.

4. Determina una opción que entregue el mismo resultado que el código a continuación:

1 / 1 punto

```
4 while a > 1:
5     a = a - 1
6     b2 = b
7     r2 = 0
8     while b2 > 0:
9         r2 = r2 + r
10        b2 = b2 - 1
11    r = r2
12 print(r)
```

- ☐ a \* b
- ☒ b \*\* a
- ☐ a \*\* b
- ☐ a / b

☒ Correcto

Correcto. Este programa tiene un doble ciclo while: el primer ciclo while hace que el código se repita *a-1* veces, dado que va desde *a* hasta que *a=1* (en este caso, 3). Luego, el segundo ciclo se repite *b* veces, dado que *b2* copia el valor de *b* y lo va reduciendo de *a* 1 (*b2 = b2-1*). La variable *r2* se usa para sumar *b2* veces el valor de *r*, y luego ese valor se guarda en la variable *r*, actualizándolo. Entonces, se repite 3 veces el cálculo de *b\*r*, pero cada vez el valor de *r* es distinto: la primera vez (cuando *a = 4*), se calcula *b\*r*, que es 9. La siguiente vez (*a = 3*), se calcula *b\*r*, pero ahora *r* vale 9, por lo que *b\*r* es 27. La siguiente vez (*a = 2*), se calcula *b\*r*, pero ahora *r* vale 27, por lo que *b\*r* es 81. Es decir, estamos calculando la potencia *b\*\*a*.

5. ¿Cuál es el valor de *a* tras la ejecución de este programa?

1 / 1 punto

```
1 a = 3
2 for i in range(2, 3):
3     a = a * i
```

- ☐ 18  
☐ 3  
☐ 2  
☒ 6

☒ Correcto  
Correcto. `range(2,3)` genera únicamente el número 2, por lo que  $a = 3 \cdot 2 = 6$

6. ¿Qué debería ir en lugar de OBJECT en el siguiente código

1 / 1 punto

```
1  a = 2
2  for i in OBJECT:
3      a = i ** a
4  print(a)
```

Para que éste imprima 9

- ☐ `range(3)`  
☒ `range(1, 4)`  
☐ `range(1, 2)`  
☐ `range(1, 3)`

☒ Correcto  
Correcto. `range(2,4)` genera los números 2 y 3, por lo que  $a$  tomaría los siguientes valores:

- primera iteración:  $a = 1^{**}2$  (es decir,  $a$  será 1)
- segunda iteración:  $a = 2^{**}1$  (es decir,  $a$  será 2)
- tercera iteración:  $a = 3^{**}2$  (es decir,  $a$  será 9)

Por lo que se imprimirá 9.

7. Determine lo que debiese ir en lugar de OBJECT para que el siguiente código

1 / 1 punto

```
1  for i in OBJECT:
2      print('hola mundo')
```

Imprima 10 veces 'hola mundo'

- ☒ `range(10)`  
☐ `range(11)`  
☐ `range(0,9)`  
☐ `range(3, 12)`

☒ Correcto

Correcto. range(10) genera los números 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (no incluye el 10), por lo tanto se repetiría el código 10 veces.

8. Determine lo que imprime este programa

1 / 1 punto

```
1 a = 0
2 for i in range(3):
3     a = a + i
4 print(a)
```

- ☒ 3  
☐ 6  
☐ 5  
☐ 4

☒ Correcto

Correcto. range(3) genera los números 0, 1, 2, por lo que el for realizaría lo siguiente:

- Cuando i=0, a = 0+0, es decir a=0
- Cuando i=1, a = 0+1, es decir a=1
- Cuando i=2, a=1+2, es decir a=3

Por lo tanto, se imprimirá 3

9. ¿Qué hace el siguiente código?

1 / 1 punto

```
1 numero = 1
2 while numero <= 5:
3     print(numero, numero**2)
```

- ☐ Escribe los números del 1 al 5 ("1", "2", "3", "4", "5") y a continuación los números del 1 al 5 al cuadrado. ("1", "4", "9", "16", "25")
- ☐ Escribe los números del 1 al 5 acompañados de ese mismo número al cuadrado ("1 1", "2 4", "3 9", "4 16", "5 25")
- ☒ Escribe "1 1" infinitas veces.
- ☐ Escribe los números del 1 al 5 acompañados de ese mismo multiplicado por 2 ("1 2", "2 4", "3 6", "4 8", "5 10")

☒ Correcto

Correcto. Dado que nos olvidamos de escribir numero= numero+1, el programa se queda en un ciclo infinito escribiendo "1 1".

10. ¿Qué imprime el siguiente código?

1 / 1 punto

```
1 for i in range(1,101):
2     for j in range(1,101):
3         print(i,j)
```

- ☐ Los números del 1 al 100, y luego los números del 1 al 100 nuevamente.
- ☐ Los números del 1 al 10000 ( $100 \times 100$ )
- ☐ Por cada número del 1 al 100, imprime el mismo número desde el 1 al 100, por lo tanto imprime 1 1, 2 2, 3 3, 4 4, ..., 100 100
- ☒ Para cada número del 1 al 100, imprime los números del 1 al 100, por lo tanto imprime 1 1, 1 2, 1 3, 1 4, ..., 1 100, 2 1, 2 2, 2 3, ..., 100 100

☒ Correcto  
Correcto. Para cada valor de  $i$ ,  $j$  tomará los valores desde 1 a 100.