

1. Los programas siguientes están mal escritos. Explica, en cada caso, por qué:

```
print(";Hola! Escribe un numero: ")
n = int(input())
if(n < 0): print("Es negativo.")
print("Hemos terminado.")
```

```
print(";Hola, mundo!")
```

```
n = input()
if(n == 4)
    print("Cuatro")
```

```
for i in range(5):
    cuadrado = i ** 2
    print((i, i ** 2))
    i = i + 1
```

2. Las siguientes afirmaciones referidas al lenguaje Python, ¿son ciertas o falsas?

1. No podemos conseguir un programa que escriba por pantalla la palabra «piña».
2. piña es un nombre válido para una variable.
3. Los objetos de tipo tuple son *inmutables*.
4. Los objetos de tipo list son *inmutables*.
5. La instrucción if sirve para que un bloque de código se ejecute varias veces.
6. El siguiente fragmento de código provocará un error al ejecutar el programa: `if (False and False): print(5 / 0)`
7. La instrucción `a = 7` compara el valor de la variable a con un objeto de tipo int.

3. Considera este fragmento de código, cuyo flujo de ejecución se representa debajo, indicando las líneas que va recorriendo el intérprete:

```
1 cadena = 'Esparadrapo'
2 i = 0
3 seguimos = True
4 while seguimos:
5     letra = cadena[i]
6     print(letra)
7     if letra == 'a' or i + 1 == len(cadena):
8         seguimos = False
9     i += 1
```

1,2,3,4,5,6,7,9,4,5,6,7,9,4,5,6,7,9,4,5,6,7,8,9,4

¿Qué valor tienen, al final de la ejecución, las variables i, letra y seguimos?

4. Escribir el siguiente código con bucles while en lugar de for:

```
s = 'abcdefghijklmñopqrstuvwxyz'
for i in range(27):
    for j in range(0, i - 1):
        for k in range(j, i):
            print(s[k] + s[j] + s[i])
```

5. Se consideran el prog1.py utilizando una única bifurcación condicional:

```

_____ prog1.py _____
1 a = int(input(" Dame un numero: "))
2 b = float(input(" Dame otro numero: "))
3
4 if (a%3 == 0 and b <= 5) or (not b/2 > 1.4 and a*b != 4):
5     print(1)
6 else:
7     print(0)
_____

```

```

_____ prog2.py _____
1 a = int(input("Dame un numero: "))
2 b = float(input("Dame otro numero: "))
3
4 if ## CONDICIÓN
5     print(0)
6 else:
7     print(1)
_____

```

Completar el programa prog2.py para que tenga el mismo resultado que prog1.py, pero escribiendo la condición contraria a la que se utiliza en prog1.py y sin utilizar el operador not.

6. Supongamos que las variables B1 y B2 tienen un valor booleano. Escribir una expresión de valor booleano que sea True si y solo si exactamente una de las variables B1 y B2 es True (es decir, una es True y la otra False). La expresión debería ser False en caso contrario.

7. Escribe un programa para resolver una clásica adivinanza: 'En una granja hay A cabezas y B patas entre gallinas y conejos, ¿cuántas gallinas y cuántos conejos hay en la granja?'

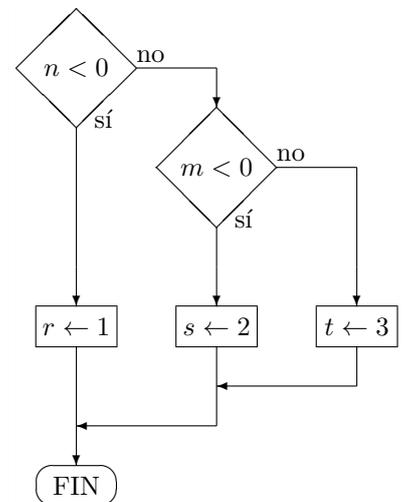
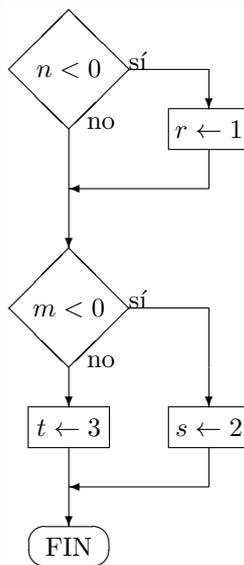
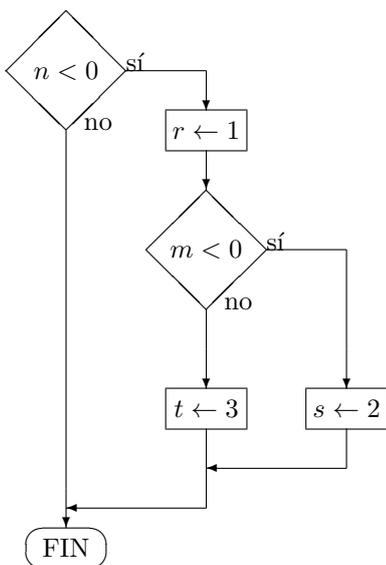
```
## AYUDA: Usar un bucle para iterar todas las posibles soluciones.
## NOTA: La adivinanza no siempre tiene solución para todos los pares A y B.
A = int(input('¿Cuántas cabezas? '))
B = int(input('¿Cuántas patas? '))
```

8. ¿Cuál de los tres diagramas representa el siguiente fragmento de código?

```

_____
if n < 0:
    r = 1
if m < 0:
    s = 2
else:
    t = 3
_____

```



Escribe en el lenguaje Python los otros dos.

9. ¿Qué se muestra por pantalla tras ejecutar el siguiente programa ?

```
B, C, D = 0, 0, 0
for A in [ 0, 1, 3, 5, 7, 9 ]:
    if A :
        if A%2 == A:
            B += 1
        else:
            C += 1
        if A%3 == 0:
            D += 1
        break
print(A) ----->
print(B) ----->
print(C) ----->
print(D) ----->
```

10. ¿ Qué se muestra por pantalla después de ejecutar estos dos programas ?

```
##PROGRAMA 1                                ##PROGRAMA 2
j = 0                                         j = 0
for i in range(2):                           while j < 2:
    while j < 2:                               for i in range(2):
        print('{0}, {1}'.format(i, j))         print('{0}, {1}'.format(i, j))
        j += 1                                 j += 1
```

11. ¿ Cúal es la salida de los tres programas siguientes? Predice el resultado sin el ordenador y comprueba después tu respuesta.

<pre>for n in range(0, 25): texto = str(n) if n % 4 == 0: texto = texto+", mult." print(texto)</pre>	<pre>for n in range(0, 25): texto = str(n) if n % 4 == 0: texto = texto+", mult." print(texto)</pre>	<pre>for n in range(0, 25): texto = str(n) if n % 4 == 0: texto = texto+", mult." print(texto)</pre>
--	--	--

12. ¿Cúal es el valor de *i*, *j* y *Lista*, después de ejecutar el siguiente código de PYTHON ?

```
j = -1; Lista=[]
for i in range(2, 5):
    Lista = [i+j] + Lista
    j = j + 1
```

13. Supongamos que *B1*, *B2*, *B3*, *B4* y *B5* son variables booleanas inicializadas. Reescribe el siguiente código para que no contenga ningún *if* anidado. El código reescrito debe ser equivalente al código dado, es decir, debe devolver la misma salida independientemente del valor de la cinco variables booleanas inicializadas.

```
if B1:
    if B2:
        print('A')
    elif B3:
        print('B')
else:
    if B4 or B5:
        print('C')
    else:
        print('D')
```

14. Cuenta la leyenda que el invento del ajedrez entusiasmó de tal modo a un rey de la India que quiso saber enseguida en qué podía ayudar al súbdito que lo había ideado. Este solo pedía arroz: un grano por la primera casilla del tablero, dos por la segunda, cuatro por la tercera, ocho por la cuarta. . . , y de este modo, doblando en cada casilla los granos de la anterior, hasta completar las sesenta y cuatro. Al rey le pareció un deseo fácil de satisfacer y se puso a la tarea de calcular la cantidad de cereal pedida para entregársela al inventor.

Escribe un programa para calcular los granos y los gramos de arroz solicitados, suponiendo que en cada gramo entran cinco granos. Escribe otro que, después de preguntar al usuario cuánto arroz hay en los graneros de palacio, calcule cuántas casillas como máximo tendría que haber tenido el tablero para que la gratificación hubiera sido factible. Por último, diseña un tercer programa que, como el anterior, calcule el mayor tamaño posible del tablero para el que se podría cumplir con el inventor, pero restringiéndose a tableros cuadrados (de una, dos, cuatro, nueve. . . casillas).

15. ¿Cuál es el resultado de la ejecución del siguiente programa? Si se produce un error describirlo.

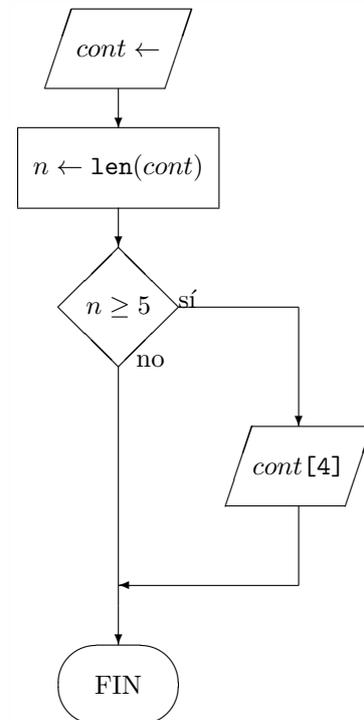
```
x = [10, 20]; y = [30, 40]
temp = x
x = y
y = temp
print(x[0], x[1])
print(y[0], y[1])
print(temp[0], temp[1])
z = x[0]
print(z)
```

16. ¿Qué resultará de ejecutar este programa ?

```
for i in range(4):
    for j in range(4):
        for k in range(i, j):
            print(i, j, k)
```

17. Escribe un diagrama de flujo que represente el programa escrito abajo, a la izquierda, y el código Python correspondiente al diagrama de flujo de la derecha. ¿Tienen el mismo resultado final estos dos programas? En caso contrario, ¿en qué se diferencian?

```
1 cont = lista
2 n = len(cont)
3 i = 0
4 while i < 4 and i < n - 1:
5     i += 1
6 print(cont[i])
```



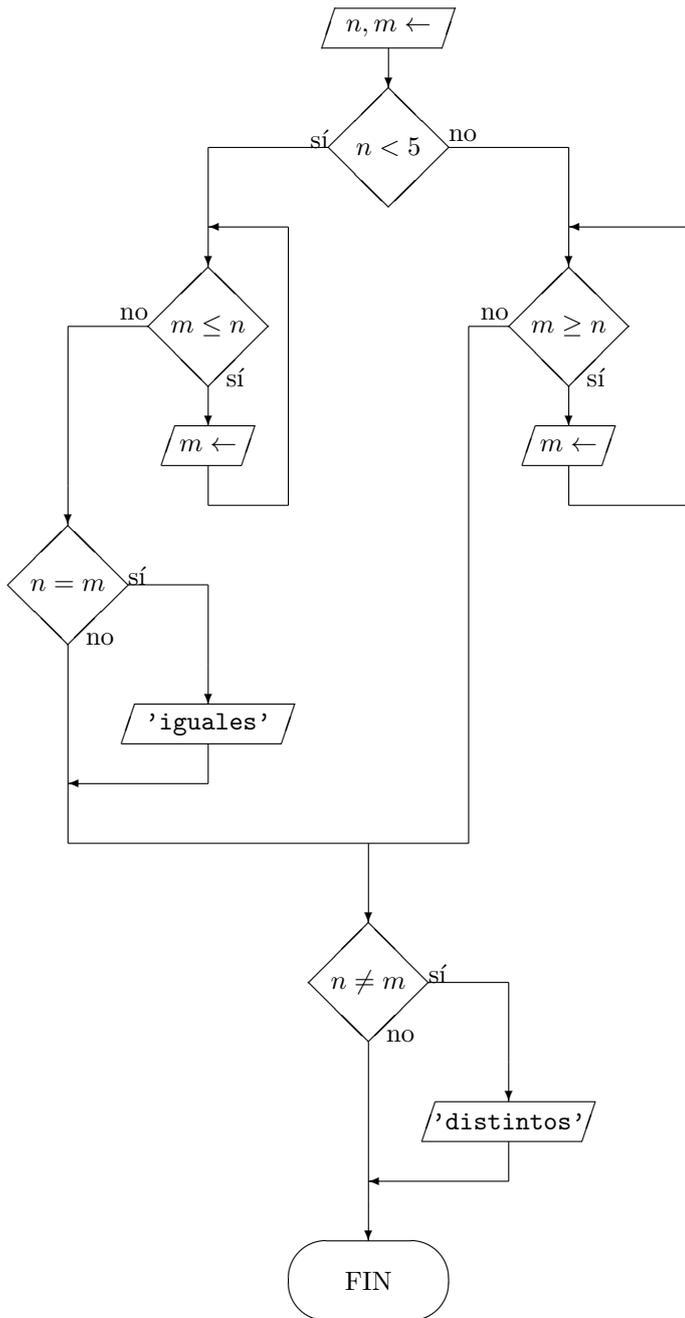
18. Implementa un programa que realice lo siguiente: pedir al usuario una cadena de caracteres y escribir la posición de la última letra *t* de la cadena, si hay alguna; o `False`, si no la hay.

19. ¿Cuál es el valor de i, j, k y L , después de ejecutar el siguiente código de PYTHON ?

```
i = 1; j = 1
L = [-1]
for k in range(1, 7, 3):
    i *= k-1
    L.append(j)
    j += i
```

20. Un palíndromo es una palabra que se lee igual de izquierda a derecha, que de derecha a izquierda. Por ejemplo: *ana, anna, salas, radar*, etc. Escribe un programa en PYTHON que decida si una palabra(= cadena de caracteres compuesta solo con letras minúsculas), es un palíndromo.

21. Este diagrama se corresponde con uno de los programas: ¿con cuál de ellos?



```
1 from random import randint
2 n = randint(0,10)
3 m = randint(0, 10)
4 if n < 5:
5     if m <= n:
6         m = randint(0, 10)
7     elif n == m:
8         print('iguales')


---



```

```
1 from random import randint
2 n = randint(0,10)
3 m = randint(0,10)
4 if n < 5:
5     while m <= n:
6         m = randint(0, 10)
7     if n == m:
8         print('iguales')


---



```

```
1 from random import randint
2 n = randint(0, 10)
3 m = randint(0,10)
4 if n < 5:
5     if m <= n:
6         m = randint(0, 10)
7 if n == m:
8     print('iguales')


---



```

Dibuja un diagrama de flujo para los otros dos.

22. Escribe dos programas en PYTHON que realicen respectivamente las siguientes dos tareas:

- Solicitar al usuario dos números n y m tales que $m > 5$, para calcular

$$\prod_{i=5}^m (n + i).$$

- Calcular el producto tensorial de dos vectores (=listas de longitud n) $\vec{u} = [a_1, a_2, \dots, a_n]$ y $\vec{v} = [b_1, b_2, \dots, b_n]$ solicitados al usuario.

Se define el producto tensorial de los vectores $\vec{u} \otimes \vec{v}$ como la lista, de longitud n , cuyos elementos son también listas de longitud n , determinadas como se describe a continuación:

$$\vec{u} \otimes \vec{v} = [[a_1 b_1, a_1 b_2, \dots, a_1 b_n], [a_2 b_1, a_2 b_2, \dots, a_2 b_n], \dots, [a_n b_1, a_n b_2, \dots, a_n b_n]]$$

Por ejemplo:

$$[2, 3] \otimes [1, 5] = [[2, 10], [3, 15]] \quad [1, 5, 0] \otimes [2, 7, 1] = [[2, 7, 1], [10, 35, 5], [0, 0, 0]]$$

23. El *doble factorial* $n!!$ de un número n se define como el producto de todos los números enteros positivos, no mayores que n y con la misma paridad de este número. Por ejemplo:

$$7!! = 1 \times 3 \times 5 \times 7, \quad 6!! = 2 \times 4 \times 6.$$

Escribe un programa para calcular el *doble factorial* de un entero positivo $n > 0$.

24. Contesta a las dos preguntas siguientes, referidas al lenguaje Python:

1. Si a y b son dos variables (ya inicializadas), las instrucciones `a==b` y `type(a)==type(b)` devuelven sendos valores *booleanos*. En otras palabras: cada una de las dos expresiones es verdadera o falsa. ¿Cuáles de las cuatro situaciones siguientes pueden presentarse y cuáles no?

	<code>a == b</code>	<code>type(a) == type(b)</code>
1	V	V
2	V	F
3	F	V
4	F	F

2. Tras ejecutarse estas tres líneas de código, ¿cuál es el valor de la variable v ?

```
1 a = 6
2 b = 8
3 v = ( ( a % b ) <= a ) and ( b < a or a != a or a != b )
```

25. ¿Cuál es la salida de los tres programas siguientes? Haz la traza de cada uno de ellos y, comprueba después tu respuesta con el ordenador.

```
i = 0
while i <= 10:
    print(i)
    i += 2
print("Acabado")
-----
```

```
i = 1
while i <= 10:
    print(i)
    i *= 0
print("Acabado")
-----
```

```
i = 1
j = -1
Lista = [2, "a"]
while i + j <= 3:
    Lista = [i * j] + Lista
    j = j + 2
print(Lista)
```