Patrones con Bucles Anidados

Nombre: _____

Fecha: _____

Puntaje: _____

1.

¿Qué es un bucle anidado?

1. Un bucle que se repite infinitamente
2. Un bucle dentro de otro bucle
3. Un bucle que solo se ejecuta una vez

2.

Si un bucle exterior se ejecuta 3 veces y un bucle interior se ejecuta 4 veces, ¿cuántas veces se ejecuta el bucle interior en total? (Responde solo el número)

Respuesta: _____

3.

Observa este código: for i in range(3): for j in range(2): print(i, j) ¿Cuál de las siguientes opciones es parte de la salida?

1. 0 2
2. 0 1
3. 1 2
4. 2 2

4.

¿Qué patrón produce el siguiente código? for fila in range(4): for columna in range(fila+1): print('*', end='') print()

1. Un cuadrado de asteriscos
2. Un triángulo rectángulo
3. Una línea de asteriscos
4. Un rombo

5.

En una tabla de multiplicar creada con bucles anidados, el bucle exterior normalmente representa las _____. (Responde con una palabra)

Respuesta: _____

6.

¿Cuál de las siguientes NO es una aplicación típica de bucles anidados?

1. Generar una tabla de multiplicar
2. Dibujar un tablero de ajedrez
3. Sumar dos números
4. Imprimir una pirámide de asteriscos

7.

¿Cuántos asteriscos imprime este código? for i in range(1,4): for j in range(i): print('*') (Responde solo el número)

Respuesta: _____

8.

¿Cuál es el orden de ejecución en un bucle anidado?

1. El bucle exterior se completa completamente, luego el interior
2. Por cada iteración del bucle exterior, el interior se ejecuta completamente
3. Ambos bucles se ejecutan al mismo tiempo

9.

Si quieres imprimir una cuadrícula de 5x5 puntos, ¿cuántos puntos se imprimen en total? (Responde solo el número)

Respuesta: _____

10.

Compara estos dos códigos: A) for i in range(4): for j in range(4): print('*', end='') print() B) for i in range(4): for j in range(i+1): print(*, end='') print() ¿Cuál es la diferencia?

1. El código A imprime un triángulo, el B un cuadrado
2. El código A imprime un cuadrado, el B un triángulo
3. Ambos imprimen el mismo patrón
4. El código A imprime un rombo, el B una línea

11.

Completa el código para que imprima un triángulo de 5 filas: for i in range(5): for j in range(_____): print('*', end='') print() (Responde la expresión del rango)

Respuesta: _____

12.

¿Cuál de los siguientes patrones NO se puede crear con bucles anidados simples (solo con range y print)?

1. Un cuadrado hueco
2. Una pirámide
3. Un tablero de ajedrez
4. Un círculo perfecto

13.

Imagina un reloj con bucle exterior para las horas (1 a 12) e interior para los minutos (0 a 59). ¿Cuántas iteraciones totales ocurren en 12 horas? (Responde solo el número)

Respuesta: _____

14.

¿Cuál es la principal ventaja de usar bucles anidados?

1. Programar de forma más concisa patrones con filas y columnas
2. Hacer que el programa sea más rápido
3. Evitar el uso de condiciones
4. Reducir el número de líneas de código siempre

15.

¿Cuántas veces se ejecuta la instrucción 'print(a)' en este código? for a in [1,2]: for b in [1,2,3]: if a == b: print(a) (Responde solo el número)

Respuesta: _____

16.

Ordena los siguientes pasos en la ejecución de un bucle anidado: (1) el bucle interior termina, (2) el bucle exterior avanza, (3) el bucle interior comienza, (4) se verifica la condición del bucle exterior. ¿Cuál es el orden correcto?

1. 4, 3, 1, 2
2. 3, 1, 2, 4
3. 1, 3, 4, 2
4. 4, 1, 2, 3

17.

En un patrón de filas y columnas, el bucle que controla el número de columnas es el bucle _____. (Responde una palabra)

Respuesta: _____

18.

¿Qué patrón genera este código? for i in range(3): for j in range(3-i): print(' ', end='') for k in range(2*i+1): print('*', end='') print()

1. Triángulo invertido
2. Cuadrado
3. Cuadrado hueco
4. Pirámide

19.

Completa: En un bucle anidado, el bucle interior completa todas sus iteraciones por cada _____ iteración del bucle exterior. (Responde una palabra)

Respuesta: _____

20.

¿Cuál de los siguientes códigos imprime los números del 1 al 9 en una cuadrícula de 3x3?

1. for i in range(3): for j in range(3): print(j+1, end=' ') print()
2. for i in range(3): for j in range(3): print(i+1, end=' ') print()
3. for i in range(3): for j in range(3): print(3*i+j+1, end=' ') print()
4. for i in range(3): for j in range(3): print(i+j+1, end=' ') print()