

## Nivel 1: Bucles Básicos

1. **Suma de Números:** Escribe un programa que calcule la suma de todos los números del 1 al 100 utilizando un bucle `for` o `while`.
2. **Números Pares e Impares:** Crea un programa que imprima todos los números pares del 1 al 50 y luego todos los números impares del 51 al 100.
3. **Tabla de Multiplicar:** Diseña un programa que solicite al usuario un número y muestre su tabla de multiplicar del 1 al 10.
4. **Factorial:** Escribe una función que calcule el factorial de un número entero positivo dado.
5. **Contador de Vocales:** Crea un programa que solicite al usuario una cadena de texto y cuente el número de vocales que contiene.

## Nivel 2: Bucles Anidados y Patrones

6. **Triángulo de Asteriscos:** Diseña un programa que imprima un triángulo de asteriscos como el siguiente:  
7. \*  
8. \*\*  
9. \*\*\*  
10. \*\*\*\*  
11. \*\*\*\*\*
12. **Cuadrado de Números:** Escribe un programa que imprima un cuadrado de números del 1 al 5, como este:  
13. 1 2 3 4 5  
14. 1 2 3 4 5  
15. 1 2 3 4 5  
16. 1 2 3 4 5  
17. 1 2 3 4 5
18. **Números Primos:** Crea un programa que imprima todos los números primos del 1 al 100.
19. **Búsqueda en Matriz:** Diseña un programa que busque un número específico en una matriz de números enteros.
20. **Suma de Diagonales:** Escribe un programa que calcule la suma de las diagonales principal y secundaria de una matriz cuadrada.

## Nivel 3: Funciones y Estrategias

11. **Calculadora Básica:** Crea un programa que simule una calculadora básica, permitiendo al usuario realizar operaciones de suma, resta, multiplicación y división. Implementa cada operación como una función separada.
12. **Conversión de Temperatura:** Escribe funciones que conviertan temperaturas de Celsius a Fahrenheit y viceversa.
13. **Palíndromo:** Diseña una función que determine si una cadena de texto es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).<sup>1</sup>
14. **Ordenamiento de Burbuja:** Implementa el algoritmo de ordenamiento de burbuja para ordenar una lista de números enteros de forma ascendente.
15. **Búsqueda Binaria:** Escribe una función que realice una búsqueda binaria en una lista ordenada de números enteros.

16. **Detección de Anagramas:** Crea una función que determine si dos cadenas de texto son anagramas (contienen las mismas letras en diferente orden).
17. **Sucesión de Fibonacci:** Diseña una función que genere los primeros  $n$  números de la sucesión de Fibonacci.
18. **Validación de Email:** Escribe una función que valide si una dirección de correo electrónico tiene un formato válido.
19. **Análisis de Texto:** Crea un programa que analice un texto dado y cuente la frecuencia de cada palabra.
20. **Juego de Adivinanza:** Diseña un juego en el que el ordenador elige un número aleatorio y el usuario tiene que adivinarlo. Proporciona pistas como "más alto" o "más bajo" según la suposición del usuario.

### Estrategias Generales

- **Divide y Vencerás:** Aborda problemas complejos dividiéndolos en subproblemas más pequeños y manejables.
- **Pensamiento Algorítmico:** Desarrolla la habilidad de descomponer un problema en pasos lógicos y secuenciales.
- **Práctica Constante:** La clave para dominar los algoritmos es la práctica regular. ¡No te rindas ante los desafíos!
- **Depuración:** Aprende a identificar y corregir errores en tu código de manera eficiente.
- **Optimización:** Busca formas de mejorar la eficiencia de tus algoritmos, tanto en tiempo como en espacio.