Prueba Diagnóstica de Programación

Abra un notebook en Google Colaboratory. Emplee un título nivel 2 en cada sección, pegue cada uno de los ejercicios y desarróllelo.

# Parte 1: Estructuras de Decisión (10 ejercicios)

1. Escribe un programa que reciba un número e indique si es positivo.
2. Escribe un programa que reciba una edad y determine si la persona es mayor de edad.
3. Escribe un programa que reciba dos números e imprima el mayor.
4. Escribe un programa que reciba una calificación (0-100) e indique si es Aprobado (>=60) o Reprobado.
5. Escribe un programa que determine si un número es par o impar.
6. Escribe un programa que reciba tres números e imprima el mayor de los tres.
7. Escribe un programa que reciba un año e indique si es bisiesto.
8. Escribe un programa que reciba una contraseña e indique si cumple con las siguientes reglas: más de 8 caracteres, contiene al menos un número y una letra mayúscula.
9. Escribe un programa que lea un número e imprima “Fizz” si es divisible por 3, “Buzz” si es divisible por 5 y “FizzBuzz” si es divisible por ambos.
10. Escribe un programa que simule un cajero automático: recibe el monto a retirar e imprime si la operación es posible en base al saldo disponible.

# Parte 2: Estructuras de Repetición (10 ejercicios)

1. Escribe un programa que imprima los números del 1 al 10.
2. Escribe un programa que sume los números del 1 al 100.
3. Escribe un programa que imprima los múltiplos de 3 entre 1 y 30.
4. Escribe un programa que lea un número e imprima su tabla de multiplicar del 1 al 10.
5. Escribe un programa que pida 5 números al usuario y calcule el promedio.
6. Escribe un programa que cuente cuántos números pares hay entre 1 y un número ingresado.
7. Escribe un programa que solicite contraseñas hasta que el usuario ingrese una válida, formada por su apellido y la longitud de su apellido (“Espinosa8”, por ejemplo).
8. Escribe un programa que imprima los primeros n números de la serie de Fibonacci.
9. Escribe un programa que determine si un número ingresado es primo.
10. Escribe un programa que pida una cantidad y luego ingrese esa cantidad de números, imprima cuántos son positivos, negativos y ceros.

# Parte 3: Funciones y Tipos de Datos (20 ejercicios)

1. Crea una función que reciba un número y retorne su cuadrado.
2. Crea una función que reciba un nombre y retorne un saludo personalizado (ese saludo debe ser aleatorio. Para ello, genere al menos 20 saludos).
3. Crea una función que reciba dos números y retorne su suma, resta, multiplicación y división.
4. Crea una función que reciba una edad y retorne si es mayor de edad.
5. Crea una función que reciba una palabra y determine si es un palíndromo.
6. Crea una función que reciba un texto y retorne cuántas vocales contiene.
7. Crea una función que reciba una lista de números y retorne la suma total.
8. Crea una función que reciba una lista y retorne el mayor y el menor elemento.
9. Crea una función que reciba una lista y retorne una nueva lista con los elementos en orden inverso.
10. Crea una función que reciba una lista de nombres y retorne cuántos nombres comienzan con vocal.
11. Crea una función que reciba una lista y retorne otra lista sin elementos repetidos.
12. Crea una función que reciba una lista de números y retorne solo los que son primos.
13. Crea una función que reciba un diccionario con claves 'nombre' y 'edad', y retorne un mensaje con esos datos.
14. Crea una función que reciba un diccionario de productos (clave: nombre, valor: precio) y retorne el total.
15. Crea una función que reciba un diccionario de estudiantes (clave: nombre, valor: nota) y retorne el promedio.
16. Crea una función que reciba un diccionario y un nombre, y retorne la nota si existe o 'no encontrado'.
17. Crea una función que reciba un diccionario y retorne el nombre del estudiante con mayor nota.
18. Crea una función que reciba un diccionario de productos y retorne los productos con precios mayores a un valor dado.
19. Crea una función que reciba un diccionario de empleados (clave: nombre, valor: lista de horas trabajadas) y retorne el total de horas por empleado.
20. Crea una función que reciba un diccionario anidado con datos de estudiantes (nombre  {'nota': x, 'edad': y}) y retorne el promedio de notas de los mayores de edad.