TP8 INGA GONZALO

1. Identifica si el problema requiere iteración conocida (for) o indefinida (while).
2. Declara y define las variables necesarias, como contadores o acumuladores.
3. Diseña una condición clara para detener el ciclo.
4. Si el enunciado parece ambiguo, divídelo en partes para comprender mejor sus requisitos.
5. Prueba con valores simples para asegurarte de que la lógica del ciclo funciona como se espera.

Por cada enunciado, indica:

1. **Si usarías un while o un for.**
2. **La condición inicial.**
3. **La condición de parada.**
4. **Si necesitas un acumulador, contador o decrementador y para qué.**

8.1) Sumar los números del 1 al 10.

1. Usaria un ciclo For
2. Mi condición inicial seria que el acumulador inicie en 0 junto con mi contador
3. Cuando mi contador llegue a 11 se parará. (ya que necesito que el ciclo for se detenga luego de sumar el 10)
4. Lo necesito ya que a medida que estoy avanzando mi contador aumenta en 1, cambiándome el número, y mi acumulador va sumando el numero anterior y actualizando mi valor total de la suma.

8.2) Mostrar los primeros 10 números de la serie de Fibonacci.

1. Usaria un ciclo for
2. Se definen las dos primeras variables de Fibonacci: a = 0 y b = 1. Se inicializa un contador en 0 para controlar el número de iteraciones
3. Cuando mi contador llegue a 11 se parará. (ya que necesito que el ciclo for se detenga luego de sumar el 10)
4. Se usan dos variables (a y b) para ir generando los números de la serie. No necesito un acumulador ya que voy actualizando los valores que voy a mostrar.

8.3) Encontrar cuántos días faltan para alcanzar una meta de ahorro semanal de $500, partiendo de $0 y ahorrando $50 por día.

1. Usaría un ciclo while, y pongo de condición cuando se alcance la meta de $500, sino es así el ciclo continua.
2. Se inicia con un acumulador (ahorro = 0). Se define un contador de días (dias = 0).
3. El ciclo debe ejecutarse hasta que el ahorro alcance o supere $500.
4. Voy a necesitar un acumulador (ahorro) para llevar la cuenta del dinero ahorrado. Contador (dias) para saber cuántos días tomará alcanzar la meta.

8.4) Contar cuántas veces aparece un carácter en una cadena de texto.

1. Utilizo un ciclo for, ya que voy a necesitar repetir el ciclo hasta que se hayan recorrido todos los caracteres
2. Se define un contador (contador = 0) para contar las apariciones del carácter y se necesita una cadena de texto y un carácter a buscar.
3. Cuando se recorre toda la cadena de texto.
4. Se necesita un contador (contador) para contar las veces que aparece el carácter buscado.

8.5) Calcular cuántas filas completas pueden formarse con 150 sillas distribuidas en grupos de 10.

1. No usaría ningún ciclo ya que es una simple cuenta matemática
2. Se define el total de sillas: sillas = 150 y defino cuántas sillas hay por fila: sillasFila = 10.
3. No se necesita un ciclo, ya que el cálculo se hace en un solo paso con 150 // 10.
4. Se necesita solo una variable filasCompletas para almacenar el resultado.

8.6) Imprimir todos los números ímpares entre 1 y 20.

1. Se usa un for, ya que conocemos el rango de números a recorrer (1 a 20).
2. Inicio mi contador en 0.
3. El ciclo se detiene cuando se supera el 20.
4. No se necesita un acumulador, solo un contador, la lógica se puede hacer de dos formas, inicio en 1 y voy sumando de 2 en dos, de esta manera me mostrara 1,3,5,7,9,11. O divido entre 2 y si tengo resto, es porque es impar.

8.7) Determinar si un número ingresado por el usuario es perfecto.

1. Necesito un ciclo for ya que tengo que ver si mi número cumple la condición de ser perfecto (suma de sus divisores propios (excluyendo el propio número) es igual al número ingresado).
2. Se pide al usuario un número n. Se define un acumulador (sumaDivisores = 0) para sumar los divisores propios de n.
3. El ciclo termina cuando se recorren todos los números.
4. Se necesita un acumulador (sumaDivisores)

8.8) Calcular cuántos pasos son necesarios para reducir un número a 0 restando 5 en cada iteración.

1. Uso un ciclo while, ya que no sé cuántas iteraciones necesito.
2. Se pide al usuario un número n. Se define un contador (pasos = 0) para contar cuántas veces restamos 5.
3. Cuando n sea menor o igual a 0, se detiene el ciclo.
4. Se necesita un contador (pasos) para contar las iteraciones yun decrementador (n -= 5) para restarle 5 en cada paso.

8.9) Mostrar las tablas de multiplicar del 1 al 5.

1. Utilizo un ciclo for, ya que tengo mi rango de números. Voy a necesitar dos, uno para la tabla y otro para la multiplicación
2. Se define un primer for para recorrer del 1 al 5 (tablas). Se define un segundo for dentro del primero, para recorrer del 1 al 10 (cada multiplicación).
3. El primer for termina cuando llega a 5. El segundo for termina cuando llega a 10.

4. Solo usamos contadores en los for

8.10) Verificar si una palabra es un palíndromo comparando sus letras desde los extremos hacia el centro.

1. Usare un while, ya que debemos comparar las letras desde los extremos hacia el centro, y no sé cuántas veces se repetirá el ciclo exactamente (dependerá de la longitud de la palabra).
2. Se pide al usuario una palabra. Definimos dos índices: uno al principio (inicio = 0) y otro al final.
3. El ciclo se detiene cuando los índices inicio y fin se cruzan (es decir, cuando inicio >= fin).
4. No se necesita un acumulador, pero sí decrementamos fin y aumentamos inicio en cada iteración para comparar las letras desde los extremos hacia el centro.