CAJA BLANCA

CHRISTIAN MILLÁN SORIA

1º DAW TARDE

---

1. Si tenemos el siguiente algoritmo, que averigua si un número es primo o no:

```
algoritmo primos
  escribir "Numero: "
  leer num

  cont<-2;
  swPrimo<-'V'

  mientras((cont<=(num/2))y(swPrimo == 'V'))hacer
    si((num mod cont)==0)entonces
      swPrimo<-F
    fin si

    cont<-cont+1
  fin mientras

  si(swPrimo=='V')entonces
    escribir num, " SI es primo"
  sino
    escribir num, " NO es primo"
  fin si
fin algoritmo
```

Indica varios casos de prueba posibles, así como algún procedimiento de prueba.

Algunos casos de prueba posibles podrían ser, por ejemplo:

- Caso de prueba válido: Ingresar el número 2, que es el número primo más pequeño.
- Caso de prueba válido: Ingresar el número 7, que es un número primo.
- Caso de prueba inválido: Ingresar el número 10, que no es un número primo.
- Caso de prueba válido: Ingresar el número 23, que es un número primo.
- Caso de prueba válido: Ingresar el número 29, que es un número primo.

Un posible procedimiento de prueba para verificar que el argoritmo funciona correctamente sería el siguiente (por pasos):

- Paso 1: Ingresar un número primo como entrada, por ejemplo, 7.
- Paso 2: Ejecutar el algoritmo.
- Paso 3: Verificar que el algoritmo indique que el número ingresado es un número primo.
- Paso 4: Repetir los pasos 1 a 3 con otros números primos para asegurarse de que el algoritmo los identifique correctamente.

- Paso 5: Ingresar un número que no sea primo, por ejemplo, 10.
- Paso 6: Ejecutar el algoritmo.
- Paso 7: Verificar que el algoritmo indique que el número ingresado no es un número primo.
- Paso 8: Repetir los pasos 5 a 7 con otros números que no sean primos para asegurarse de que el algoritmo los identifique correctamente.
- Paso 9: Asegurarse de que el algoritmo no tenga errores de lógica para cualquier número de entrada mediante el análisis del código fuente y la verificación de los cálculos realizados en el código.