

Tarea no evaluable.

Diseña el algoritmo mediante el diagrama de flujo y luego codifica en Python para los siguientes casos:

1. Crea un programa que escriba "Comienzo a aprender".
2. Crea un programa que escriba la frase "El resultado es", y luego, muestre el resultado de dividir 37 entre 5. 3.
3. Crea un programa que escriba el resultado de multiplicar los dos números que introduzca el usuario por teclado.
4. Crea un programa que pida un número al usuario y diga si es positivo. (Programación estructurada Alternativa simple)
5. Crea un programa que pida un número al usuario y diga si es positivo o negativo. (Programación estructurada Alternativa compuesta)
6. Crea un programa que pida un número al usuario y diga si es positivo, negativo o cero. (Programación estructurada Alternativa compuesta)
7. Haz un programa que pida al usuario dos números y diga cuántos de ellos son positivos. (Programación estructurada Alternativa)
8. Crea un programa que pida al usuario dos números y muestre su división si el segundo no es cero, o un mensaje de aviso en caso contrario. (Programación estructurada Alternativa compuesta)
9. Prepara un programa que pida al usuario tres números y diga cuál es el mayor de los tres. (Programación estructurada Alternativa compuesta)
10. Crea un programa que pida al usuario una contraseña, de forma repetitiva mientras que no introduzca "1234". Cuando finalmente escriba la contraseña correcta, se le dirá "Bienvenido" y terminará el programa. (Programación estructurada Repeticiones)
11. Haz un programa que permita calcular la suma de dos números. Pedirá dos números al usuario y mostrará su suma, volviendo a repetir hasta que ambos números introducidos sean 0. (Programación estructurada Repeticiones)
12. Crea un programa que genere dos números al azar entre el 0 y el 100, y pida al usuario que calcule e introduzca su suma. Si la respuesta no es correcta, deberá volver a pedirla tantas veces como sea necesario hasta que el usuario acierte. Pista: para generar un número al azar del 0 al 100 puedes hacer `número <- AZAR(101)` (Programación estructurada Repeticiones)