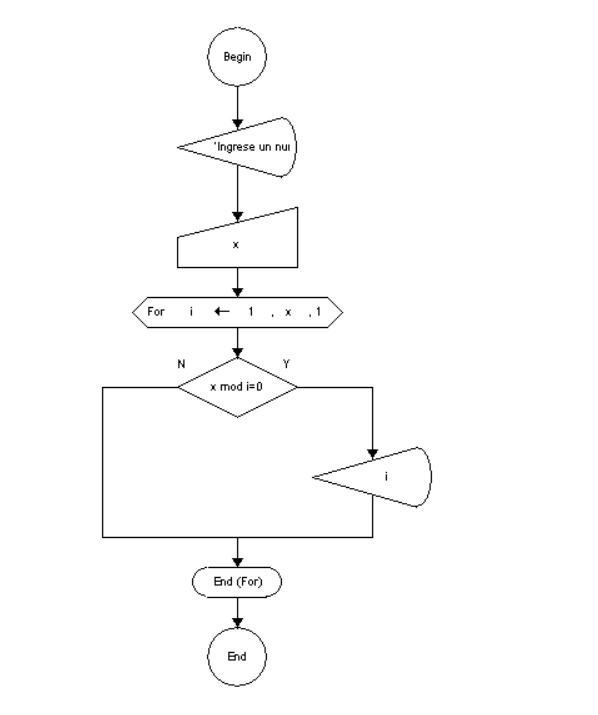
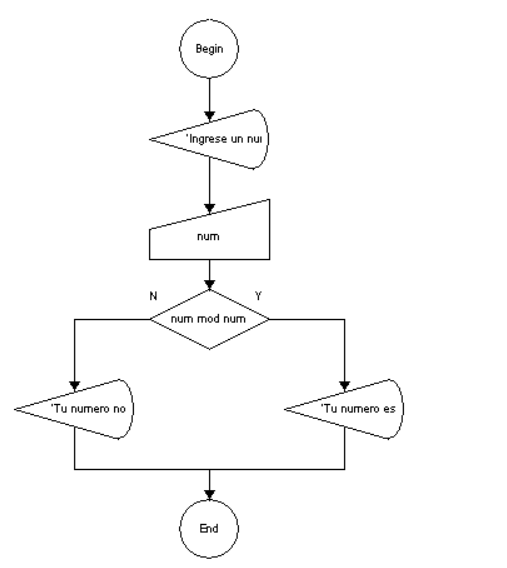
1. **Determinar los divisores de un número introducido por teclado**

****

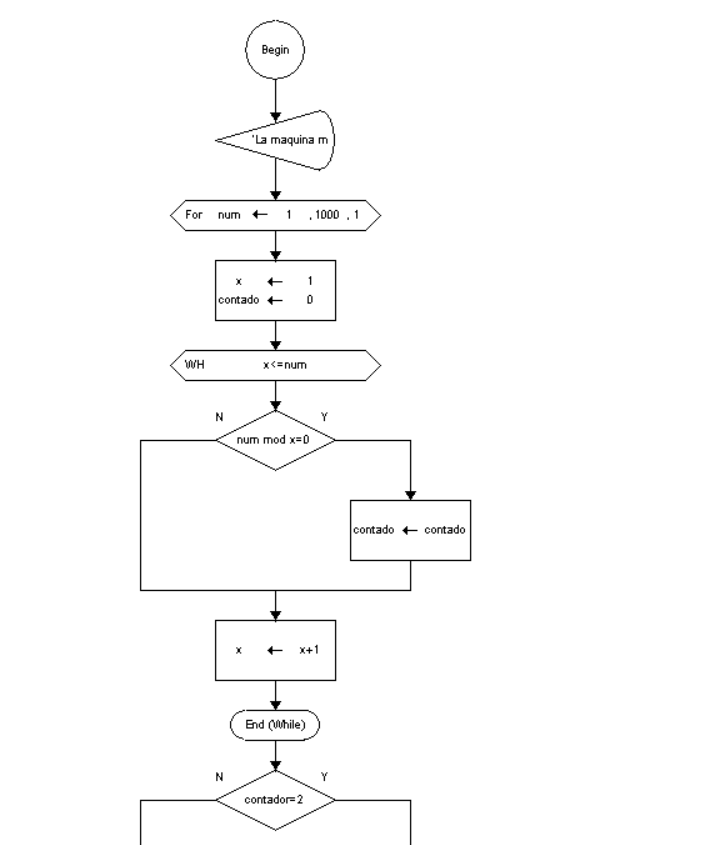
1. **Determinar si un número es o no es primo**

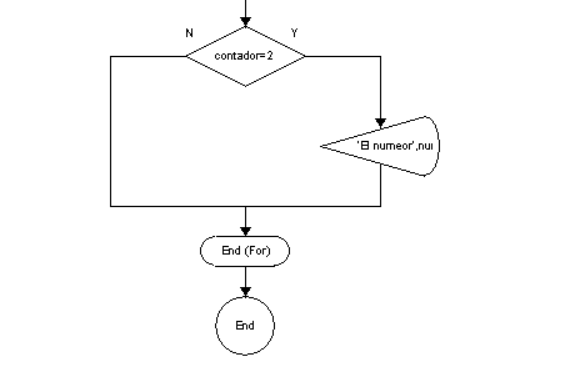
****

**3. Determinar si un número es o no es perfecto. Un numero es perfecto si la suma de sus divisores sin incluir el propio número es igual a este**

**4. Determinar cuales y cuantos números perfectos hay entre 1 y 1000?**

**5. ¿Cuáles y cuántos son los números primos comprendidos entre 1 y 1000?**

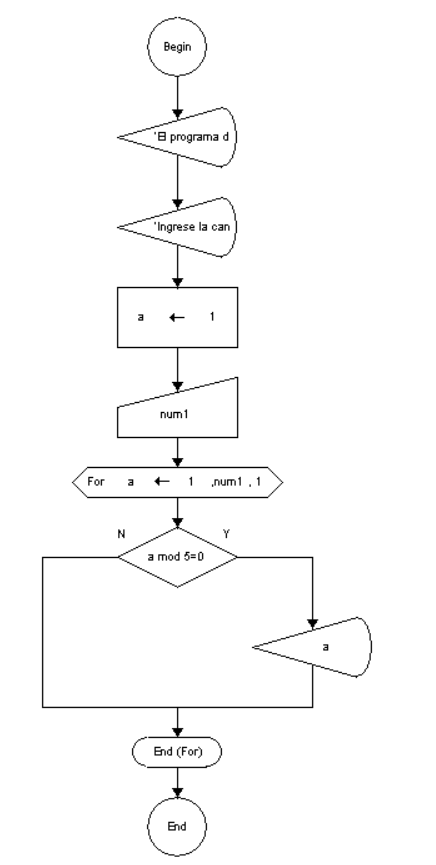
****

****

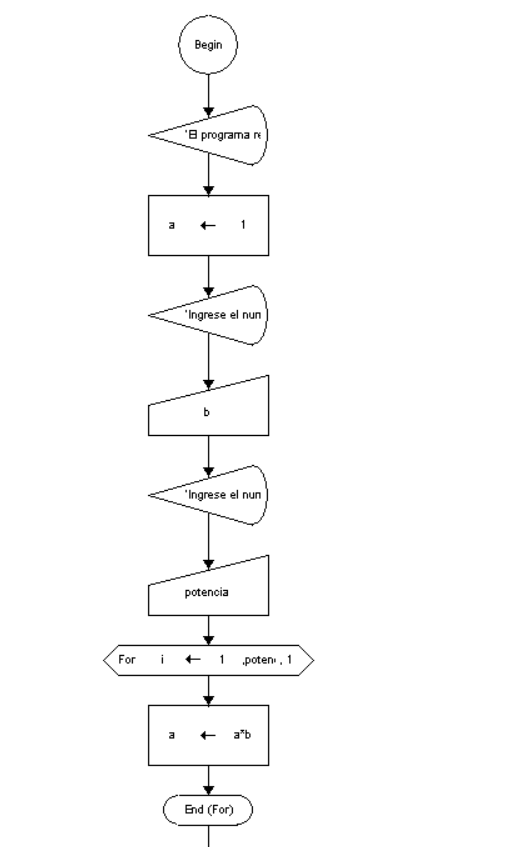
**6. Calcular el máximo de números positivos introducidos por teclado, sabiendo que metemos números hasta que introduzcamos uno negativo. El negativo no cuenta.**

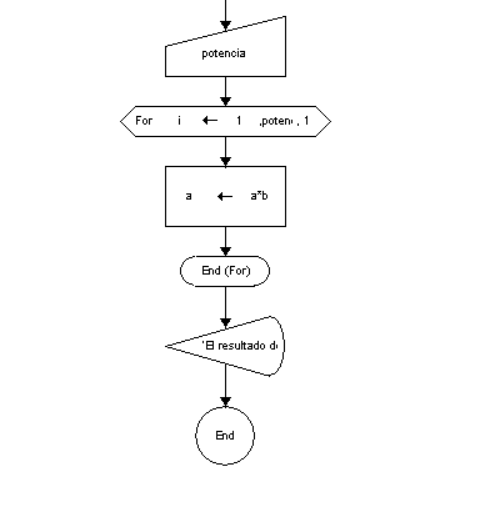
**7. Encontrar un número natural n más pequeño tal que la suma de los n primeros números naturales (1 + 2 + 3 + 4…..) exceda de una cantidad (máximo) introducida por el teclado. Es decir cuantos números de la serie de los naturales debo sumar para superar el dato máximo.**

**8. Determinar cuales son los múltiplos de 5 comprendidos entre 1 y N.**

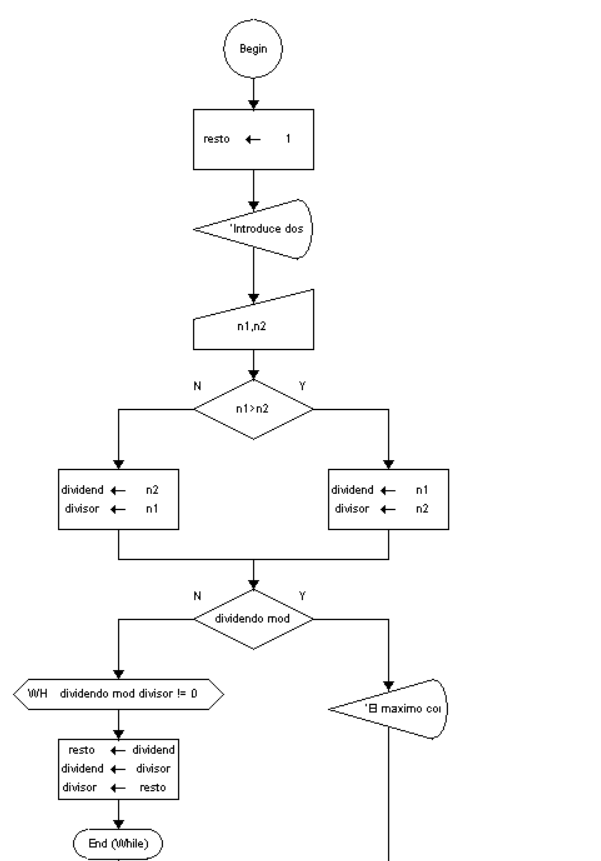
****

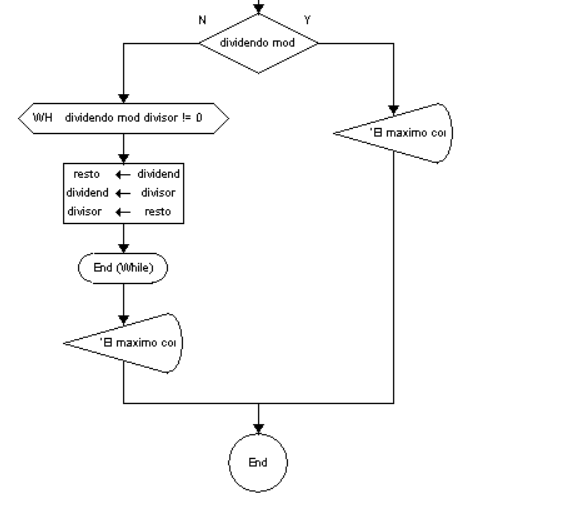
**9. Calcular la operación x n sin utilizar la función pow**

****

****

**10. Algoritmo para hallar el m.c.d de dos números m y n por el algoritmo de Euclides.**

****

****

**11. Solicitar 2 números al usuario e imprimir el cociente y el residuo del mayor en el menor sin utilizar la división ni el mod.**

**12. Escribir un programa que visualice la siguiente figura, utilizando ciclos. El usuario decide cuantas líneas quiere imprimir**

**13. Solicitar al usuario un número de hasta 9 dígitos e imprimirlo en orden contrario. Ej. digito 6754 imprimo 4576**

**14. Calcular todos los números de 3 cifras tales que la suma de los cubos de las cifras es igual al valor del número.**

**15. Diseñar e implementar un programa que solicite a su usuario un valor no negativo n y visualice la siguiente salida: 1 2 3 ........ n-1 n 1 2 3 ........ n-1 ......... 1 2 3 1 2 1**