

Algunos ejercicios resueltos de la Unidad 1

Ejercicios de Programación Estructurada (números 21 a 55)

22. Determinar si un número introducido por teclado es positivo o negativo

Algoritmo positivo_negativo

Variables

n es entero

Inicio

escribir("Introduce un número: ")

leer(n)

si (n > 0)

inicio

escribir("El número es positivo")

fin

si (n < 0)

inicio

escribir("El número es negativo")

fin

si (n == 0)

inicio

escribir("El número no es positivo ni negativo")

fin

fin

Solución alternativa:

Algoritmo positivo_negativo_2

Variables

n es entero

Inicio

escribir("Introduce un número: ")

leer(n)

si (n > 0)

inicio

escribir("El número es positivo")

fin

si_no

inicio

si (n < 0)

inicio

escribir("El número es negativo")

fin

si_no

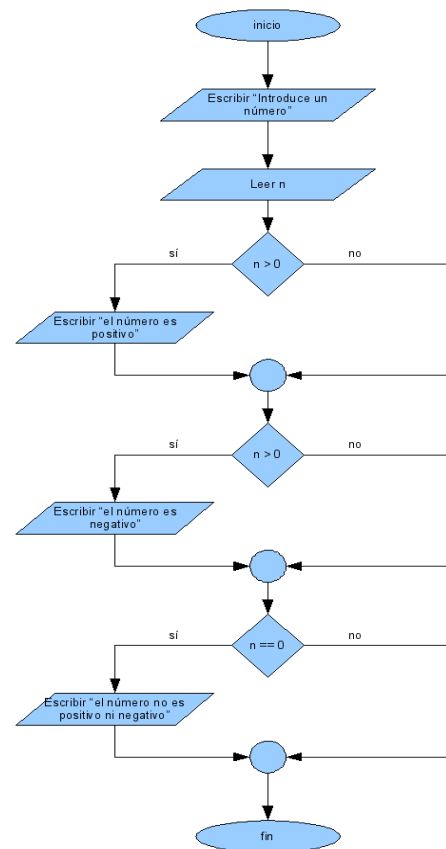
inicio

escribir("El número no es positivo ni negativo")

fin

fin

fin



23. Calcular la raíz cuadrada de un número introducido por teclado. Hay que tener la precaución de comprobar que el número sea positivo.

```
Algoritmo raiz_cuadrada
Variables
    n es entero
    r es real
inicio
    escribir("Introduce un número: ")
    leer(n)
    si (n >= 0)
        inicio
            r = raiz(n)
            escribir("La raíz es: ", r)
        fin
    si_no
        inicio
            escribir("Error: No existe la raíz de un número negativo")
        fin
fin
```

24. Dados dos números A y B por teclado, restar el mayor menos el menor.

```
Algoritmo restar_dos_números
Variables
    A, B, C son enteros
Inicio
    Escribir("introduce el primer número")
    Leer(A)
    Escribir("introduce el segundo número")
    Leer(B)

    Si (A > B)
        inicio
            C = A - B
        fin
    Si_no
        inicio
            C = B - A
        fin

    Escribir("El resultado es: ", C)
Fin
```

25. Determinar si un año es bisiesto o no. Recuerda que los años bisiestos son aquellos cuya cifra es un múltiplo de 4.

```
Algoritmo bisiestos
Variables
    A es entero
Inicio
    Escribir("Introduce el año: ")
    Leer(A)
    Si (A % 4 == 0)
        Escribir("El año es bisiesto")
    Si_no
        Escribir("El año NO es bisiesto")
Fin
```

Solución alternativa. Esta solución tiene en cuenta que los años múltiplos de 100 NO son bisiestos, a pesar de ser también múltiplos de 4. Pero, de ellos, los múltiplos que también lo son de 400 SÍ que son bisiestos.

Algoritmo bisiestos

Variables

A es entero

Inicio

 Escribir("Introduce el año: ")

 Leer(A)

 Si (A % 4 == 0)

 Inicio

 Si (A % 100 == 0)

 Inicio

 Si (A % 400 == 0) // Es múltiplo de 4, de 100 y de 400

 Inicio

 Escribir("El año es bisiesto")

 Fin

 Si_no // Es múltiplo de 4 y de 100, pero no de 400

 Inicio

 Escribir("El año NO es bisiesto")

 Fin

 Fin

 Si_no // Es múltiplo de 4 pero no de 100

 Inicio

 Escribir("El año es bisiesto")

 Fin

 Si_no // No es múltiplo de 4

 Inicio

 Escribir("El año NO es bisiesto")

 Fin

Fin

26. Averiguar si un número real introducido por teclado tiene o no parte fraccionaria

algoritmo parte_fraccionaria

variables

 x es real

 n es entero

inicio

 escribir("Introduce un número real: ")

 leer(x)

 n = trunc(x)

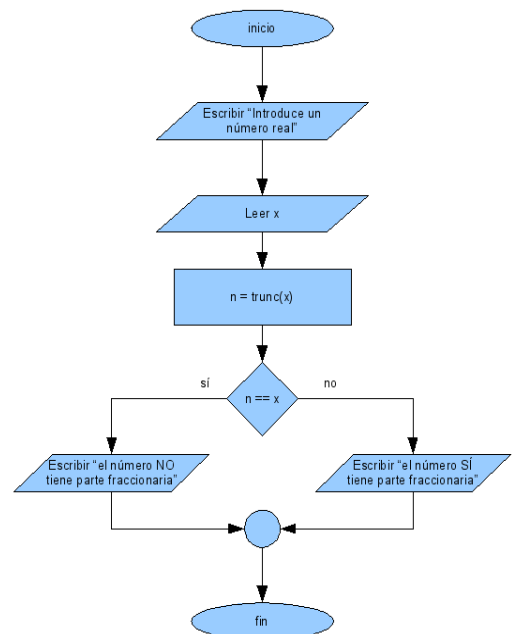
 si (x == n)

 Escribir("El número NO tiene parte fraccionaria")

 si_no

 Escribir("El número SÍ tiene parte fraccionaria")

fin



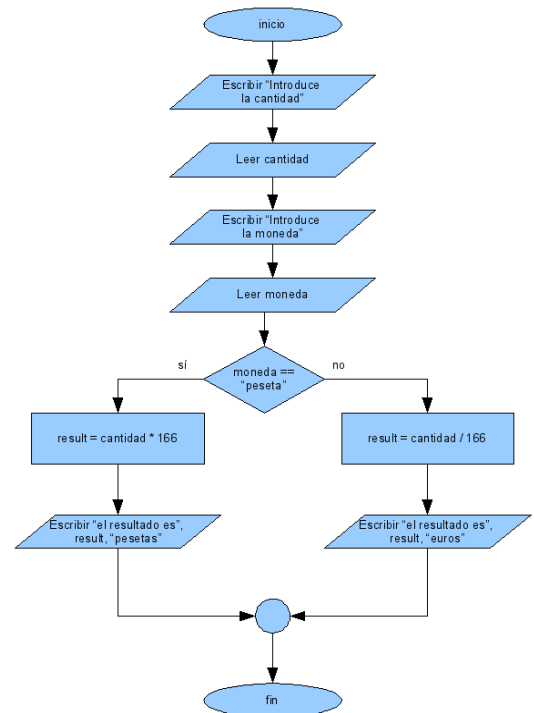
27. Leer un número real y un tipo de moneda, que puede ser "euro" o "peseta". Convertir la cantidad al tipo de moneda indicado, suponiendo que está expresada en la otra. Por ejemplo, si la cantidad es 15 y la moneda es "peseta", se supondrá que se trata de 15 € y que hay que convertirlos a pesetas y, por lo tanto, el resultado debe ser 2495.

```

algoritmo euros_y_pesetas
variables
    cantidad, result son reales
    moneda es cadena
inicio
    escribir("Introduce la cantidad: ")
    leer(cantidad)
    escribir("Introduce la moneda (euro o peseta): ")
    leer(moneda)

    si (moneda == "peseta")      // Lo convertimos a PESETAS
    inicio
        result = cantidad * 166
        escribir("El resultado es: ", result, " pesetas")
    fin
    si_no                        // Lo convertimos a EUROS
    inicio
        result = cantidad / 166
        escribir("El resultado es: ", result, " euros")
    fin
fin

```

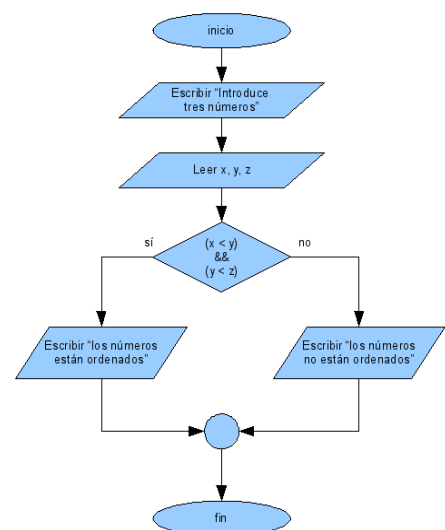


28. Leer tres números por teclado, X, Y y Z, y decidir si están ordenados de menor a mayor.

```

algoritmo numeros_ordenados
variables
    x, y, z son enteros
inicio
    escribir("Introduce tres números enteros")
    leer(x, y, z)
    si (x < y) && (y < z)
        escribir("Los números están ordenados de menor a mayor")
    si_no
        escribir("Los números no están ordenados")
fin

```



29. Leer tres números por teclado, X, Y y Z, y decidir si están ordenados de menor a mayor y, además, son *consecutivos*.

algoritmo numeros_consecutivos

variables

x, y, z son enteros

inicio

escribir("Introduce tres números enteros")

leer(x, y, z)

si (x < y) && (y < z)

inicio

si ((x + 1) == y) && ((y + 1) == z)

escribir("Los números están ordenados y son consecutivos")

si_no

escribir("Los números están ordenados pero NO son consecutivos")

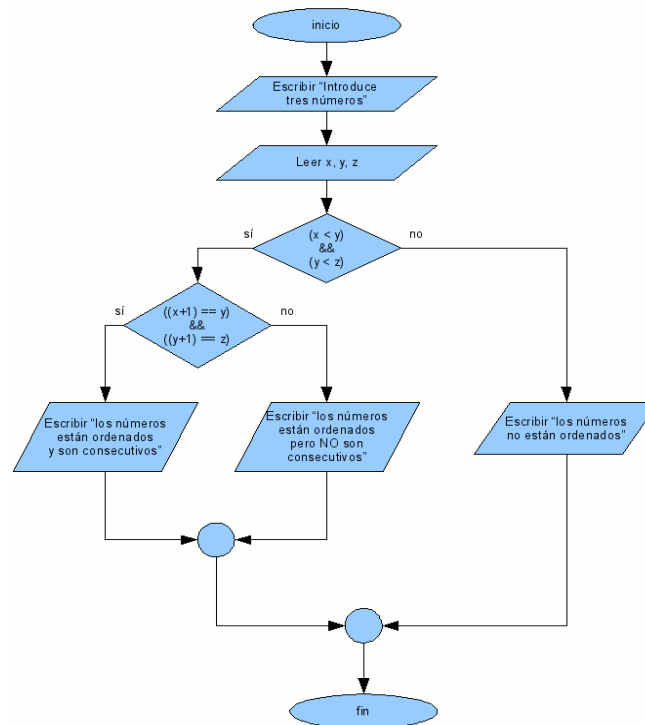
fin

si_no

escribir("Los números no están ordenados")

fin

Diagrama de flujo:



30. Determinar el número de cifras de un número entero. El algoritmo debe funcionar para números de hasta 5 cifras, considerando los negativos. Por ejemplo, si se introduce el número 5342, la respuesta del programa debe ser 4. Si se introduce -250, la respuesta debe ser 3

```
algoritmo contar_cifras
variables
    n es entero
inicio
    escribir("Introduzca un número (máximo 5 cifras)")
    leer(n)
    n = abs(n)
    si (n <= 9) entonces
        escribir("El número tiene 1 cifra")
    si (n >= 10) y (n <= 99) entonces
        escribir("El número tiene 2 cifras")
    si (n >= 100) y (n <= 999) entonces
        escribir("El número tiene 3 cifras")
    si (n >= 1000) y (n <= 9999) entonces
        escribir("El número tiene 4 cifras")
    si (n >= 10000) y (n <= 99999) entonces
        escribir("El número tiene 5 cifras")
    si (n > 99999) entonces
        escribir("Error: ha introducido un número de más de 5 cifras")
fin
```

31. Calcular las dos soluciones de una ecuación de segundo grado, del tipo $ax^2 + bx + c = 0$. Los coeficientes a, b y c deberá introducirlos el usuario a través del teclado

```
algoritmo ecuacion
variables
    a, b, c son reales
    rad es real
    x1, x2 son reales
inicio
    escribir("Introduzca los tres coeficientes de la ecuación de segundo grado");
    leer(a)
    leer(b)
    leer(c)
    rad = b^2 - 4 * a * c
    si (rad < 0) entonces
        escribir("La ecuación no tiene solución real")
    si_no
        inicio
            x1 = - b + raiz(rad) / (2 * a)
            x2 = - b - raiz(rad) / (2 * a)
            escribir("Las soluciones son: ", x1, x2)
        fin
    fin
fin
```

32. Dados tres números enteros, A, B, C, determinar cuál es el mayor, cuál el menor y cuál el mediano.

```
algoritmo mayor_mediano_menor
variables
    a, b, c son enteros
inicio
    escribir("Introduzca tres números enteros")
    leer(a)
    leer(b)
    leer(c)

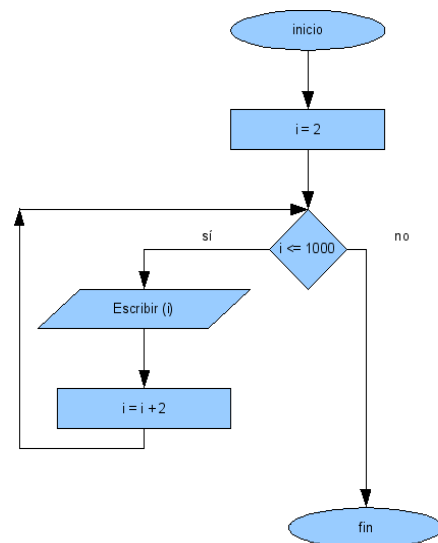
    si (a > b) y (a > c) entonces
        inicio
            si (b > c) entonces
                escribir("El mayor es", a, ", el mediano es ", b, " y el menor es ", c)
            si_no
                escribir("El mayor es", a, ", el mediano es ", c, " y el menor es ", b)
        fin
    fin

    si (b > a) y (b > c) entonces
        inicio
            si (a > c) entonces
                escribir("El mayor es", b, ", el mediano es ", a, " y el menor es ", c)
            si_no
                escribir("El mayor es", b, ", el mediano es ", c, " y el menor es ", a)
        fin
    fin

    si (c > a) y (c > b) entonces
        inicio
            si (a > b) entonces
                escribir("El mayor es", c, ", el mediano es ", a, " y el menor es ", b)
            si_no
                escribir("El mayor es", c, ", el mediano es ", b, " y el menor es ", a)
        fin
    fin
fin
```

33. Escribir todos los números pares entre 1 y 1000.

```
algoritmo pares_de_1_a_1000
variables
    i es entero
inicio
    i = 2
    mientras ( i <= 1000 )
        inicio
            escribir(i)
            i = i + 1
        fin
    fin
fin
```



34. Escribir todos los números impares entre dos números A y B introducidos por teclado. Antes habrá que comprobar cuál de los dos números A y B es mayor .

algoritmo impares_de_A_a_B

variables

a, b, i son enteros

inicio

escribir("Introduzca el valor de A: ")

leer(a)

escribir("Introduzca el valor de B: ")

leer(b)

// Establecemos el numero de inicio y de fin del bucle

si (a < b)

inicio

i = a

f = b

fin

si_no

inicio

i = b

f = a

fin

// Si el punto de inicio es par,

// lo "movemos" al siguiente número impar

si (i % 2 == 0)

i = i + 1

// Por fin, escribimos los números

mientras (i < f)

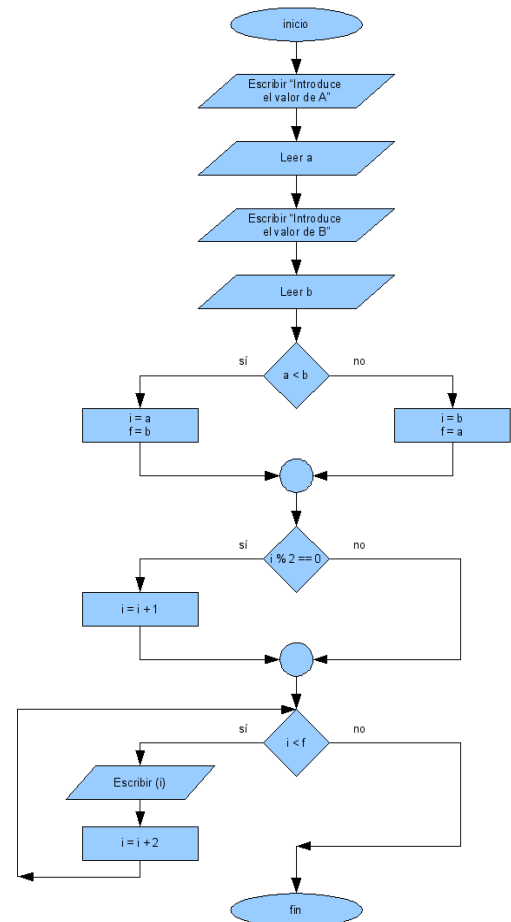
inicio

escribir(i)

i = i + 2

fin

fin

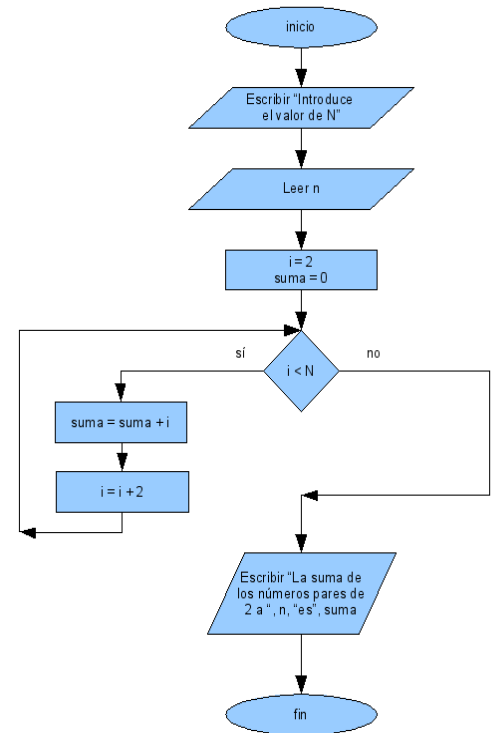


35. Calcular la suma de todos los números pares entre 1 y N. Es decir, $2 + 4 + 6 + \dots + (N-2) + N$. El valor de N se introducirá por teclado.

```

algoritmo suma_de_pares_de_1_a_1000
variables
    i, n, suma son enteros
inicio
    escribir("Introduzca el valor de N: ")
    leer(n)
    i = 2
    suma = 0
    mientras (i < N) hacer
        inicio
            suma = suma + i
            i = i + 2
        fin
    fin
    escribir("La suma de los números pares de 2 a ", n, " es ", suma)
fin

```



36. Mostrar el mensaje "¿Desea terminar? (S/N)" y leer la respuesta del usuario. Si es "S", el programa terminará.

```

algoritmo desea_terminar?
variables
    resp es carácter
inicio
    repetir
        inicio
            escribir("¿Desea terminar (S / N) ?")
            leer(resp)
        fin
    mientras (resp == 'N')
fin

```

37. Calcular el valor medio de una serie de valores enteros positivos introducidos por teclado. Para terminar de introducir valores, el usuario debe teclear un número negativo .

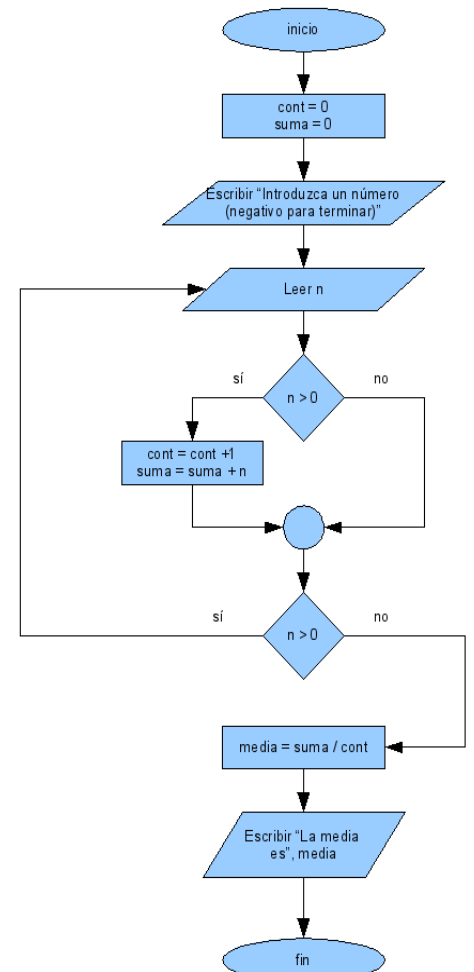
```

algoritmo calcular_media
variables
    n, cont, suma son enteros
    media es real
inicio
    cont = 0
    suma = 0

    // Introducir los números (hasta que se introduzca un negativo)
    repetir
        inicio
            escribir("Introduzca un número (negativo para terminar)")
            leer(n)
            si (n > 0)
                inicio
                    cont = cont + 1 // Llevamos la cuenta de la cantidad de números introducidos
                    suma = suma + n // Acumulamos el resultado de las sumas sucesivas
                fin
            fin
        mientras (n > 0)

    // Calcular la media y mostrar el resultado
    media = suma / cont
    escribir("La media es: ", media)
fin

```



38. El usuario de este programa será un profesor, que introducirá las notas de sus 30 alumnos de una en una. El algoritmo debe decirle cuántos suspensos y cuántos aprobados hay .

algoritmo contar_suspensos_y_aprobados
variables
 cont_susp, cont_aprob, i son enteros
 nota es real

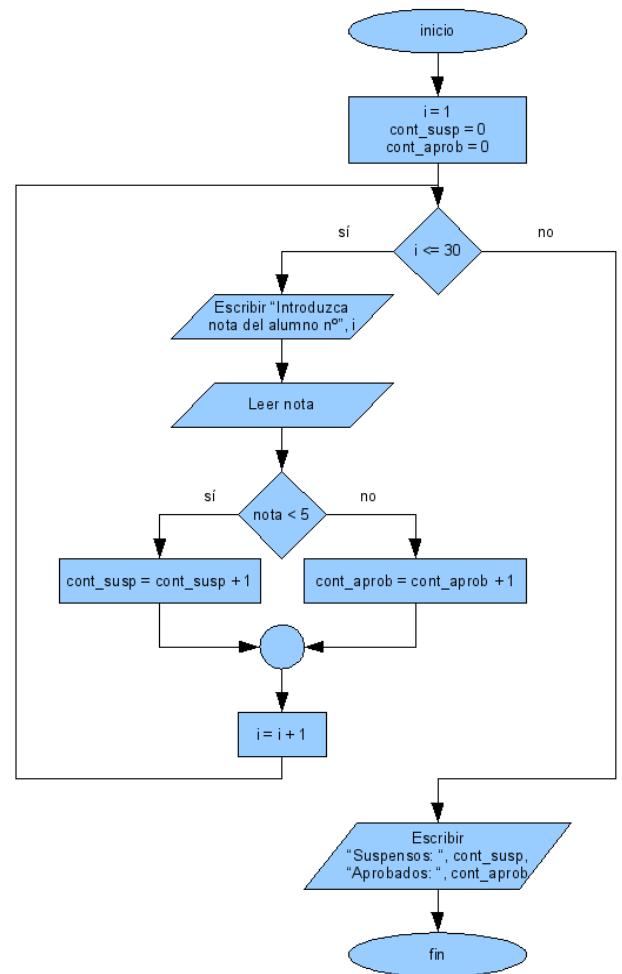
```

inicio
    i = 1
    cont_susp = 0
    cont_aprob = 0

    mientras (i <= 30)
        inicio
            escribir("Introduzca la nota del alumno nº ", i)
            leer(nota)
            si (nota < 5)
                inicio
                    // Contamos un suspenso más
                    cont_susp = cont_susp + 1
                fin
            si_no
                inicio
                    // Contamos un aprobado más
                    cont_aprob = cont_aprob + 1
                fin
            i = i + 1
        fin

    escribir("Suspensos: ", cont_susp)
    escribir("Aprobados: ", cont_aprob)
fin

```



39. Calcular el valor máximo de una serie de 10 números introducidos por teclado.

```

algoritmo máximo
variables
    n, mayor, i son enteros
inicio
    mayor = <<el número más pequeño posible>>
    para i desde 1 hasta 10 hacer
        inicio
            escribir("Introduce un número")
            leer(n)
            si (n > mayor) // Hemos encontrado un nuevo máximo
                inicio
                    mayor = n
                fin
        fin
    escribir("El máximo es: ", mayor)
fin

```

40. Generalizar el ejercicio anterior para que también se averigüe el valor mínimo y el medio.

```
algoritmo máximo_mínimo_y_media
variables
    n, mayor, menor, suma, i son enteros
    media es real
inicio
    mayor = <<el número más pequeño posible>>
    menor = <<el número más grande posible>>

    para i desde 1 hasta 10 hacer
    inicio
        escribir("Introduce un número")
        leer(n)
        si (n > mayor) // Hemos encontrado un nuevo máximo
        inicio
            mayor = n
        fin
        si (n < menor) // Hemos encontrado un nuevo mínimo
        inicio
            menor = n
        fin
        suma = suma + n // Acumulamos el resultado, para después calcular la media
    fin // fin-para

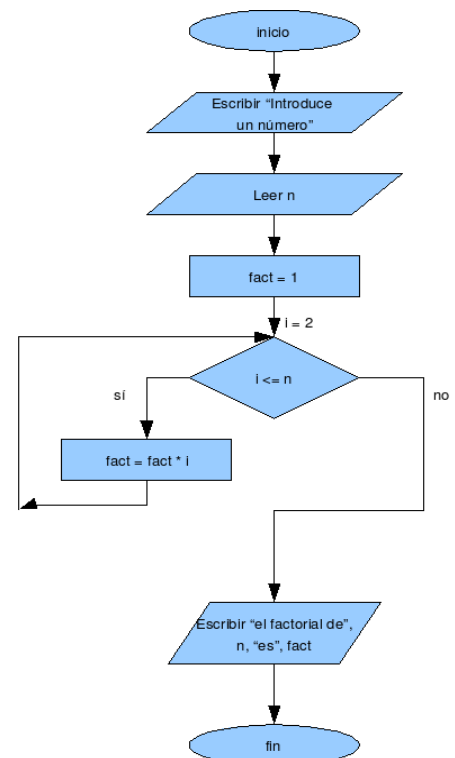
    escribir("El máximo es: ", mayor)
    escribir("El mínimo es: ", menor)
    media = suma / 10
    escribir("La media es: ", media)
fin
```

41. Calcular el factorial de un número entero N. Recuerda que el factorial de un número es el producto de ese número por todos los enteros menores que él. Por ejemplo, el factorial de 5 (simbolizado 5!) se calcula como: $5! = 5 \times 4 \times 3 \times 2 \times 1$.

```
algoritmo factorial
variables
    n, fact, i son enteros
inicio
    escribir("introduce un número")
    leer(n)
    fact = 1

    para i desde 2 hasta n hacer
    inicio
        fact = fact * i
    fin

    escribir("El factorial de ", n, " es: ", fact)
fin
```



42. Determinar si un número N introducido por teclado es o no primo. Recuerda que un número primo es aquél que sólo es divisible por sí mismo y por la unidad.

```

algoritmo primo
variables
    n, i son enteros
    es_primo es lógico
inicio
    escribir("Introduce un número")
    leer(n)

    es_primo = verdadero
    para i desde 2 hasta n-1 hacer
    inicio
        // Hemos encontrado un divisor → n no es primo
        si (n % i == 0)
        inicio
            es_primo = falso
        fin
    fin

    si (es_primo)
        escribir("El número ", n, " es primo")
    si_no
        escribir("El número ", n, " NO es primo")
fin

```

Versión alternativa. Se ejecuta más rápidamente cuando n no es primo. Si n es primo, tarda la mitad que la versión anterior, porque sólo comprueba los posibles divisores hasta $n/2$, ya que a partir de ahí es imposible encontrar ningún divisor.

```

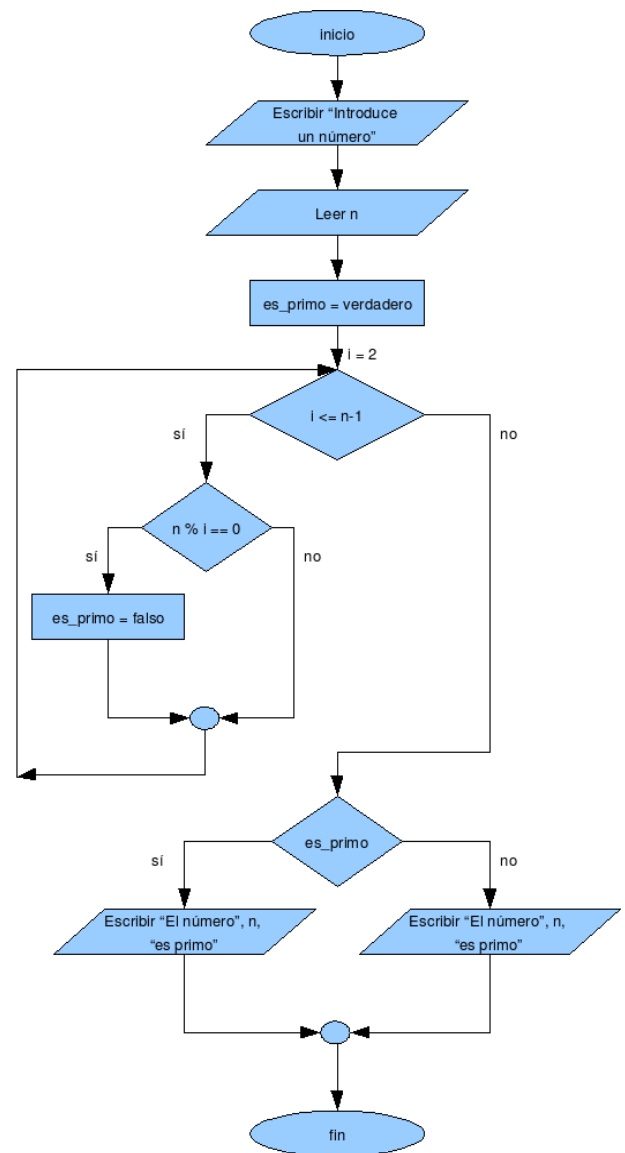
algoritmo primo2
variables
    n, i son enteros
    es_primo es lógico
inicio
    escribir("Introduce un número")
    leer(n)

    es_primo = verdadero
    i = 2

    mientras (i < n/2) && (es_primo)
    inicio
        si (n % i == 0)          // Hemos encontrado un divisor → n no es primo
        inicio
            es_primo = falso
        fin
        i = i + 1
    fin

    si (es_primo)
        escribir("El número ", n, " es primo")
    si_no
        escribir("El número ", n, " NO es primo")
fin

```



43. Generalizar el algoritmo anterior para averiguar todos los números primos que existen entre 2 y 1000.

```
algoritmo primos_de_2_a_1000
variables
    n, i son enteros
    es_primo es lógico

inicio

    para n desde 2 hasta 1000 hacer      // Vamos a mirar si cada n es primo o no
    inicio
        es_primo = verdadero
        i = 2

        mientras (i < n/2) && (es_primo)
        inicio
            si (n % i == 0)              // Hemos encontrado un divisor → n no es primo
            inicio
                es_primo = falso
            fin
            i = i + 1
        fin

        si (es_primo)
            escribir(n)

    fin      // fin-para
fin
```

44. Introducida una hora por teclado (horas, minutos y segundos), se pretende sumar un segundo a ese tiempo e imprimir en la pantalla la hora que resulta (también en forma de horas, minutos y segundos) .

```
algoritmo sumar_un_segundo
variables
    h, m, s son enteros
inicio
    escribir("Introduce la hora, los minutos y los segundos")
    leer(h, m, s)

    // Ahora vamos a sumar un segundo, a ver lo que resulta
    s = s + 1
    si (s == 60)                // Si alcanzamos 60 segundos, hay que sumar un minuto
    inicio
        s = 0
        m = m + 1
        si (m == 60)            // Si alcanzamos 60 minutos, hay que sumar una hora
        inicio
            m = 0
            h = h + 1
            si (h == 24)         // Si alcanzamos 24 horas, hay que volver al cero
            inicio
                h = 0
            fin
        fin
    fin

    fin

    escribir("La nueva hora es ", h, ":", m, ":", s)
fin
```

45. Generar combinaciones al azar para la lotería primitiva (6 números entre 1 y 49). Debes utilizar la función aleatorio(x) que vimos en los apuntes. Por ahora, no te preocupes porque los números puedan repetirse .

```

algoritmo primitiva
variables
    n, i son enteros
inicio
    escribir("Una combinación al azar para la primitiva:")
    para i desde 1 hasta 6 hacer
        inicio
            n = aleatorio(49)          //n será un número al azar de 0 a 48
            n = n + 1                  //n será un número al azar de 1 a 49
            escribir(n)
        fin
    fin
fin

```

46. Generar combinaciones al azar para la quiniela (14 valores dentro del conjunto 1, X o 2) .

```

algoritmo quiniela
variables
    n, i son enteros
inicio
    escribir("Una combinación al azar para la quiniela:")
    para i desde 1 hasta 14 hacer
        inicio
            n = aleatorio(3)          // n valdrá 0, 1 ó 2
            si (n == 0)
                escribir("X")        // Si n vale 0, escribimos una X
            si_no
                escribir(n)          // Si n vale 1 ó 2, escribimos el valor de n
        fin
    fin
fin

```

47. La calculadora. Diseñar un algoritmo que lea dos números, A y B, y un operador (mediante una variable de tipo carácter), y calcule el resultado de operar A y B con esa operación. Por ejemplo, si A = 5 y B = 2, y operación = "+", el resultado debe ser 7. El algoritmo debe seguir pidiendo números y operaciones indefinidamente, hasta que el usuario decida terminar (utilizar un valor centinela para ello) .

```

algoritmo calculadora
variables
    a, b, r son reales
    op, terminar es carácter
inicio
    repetir
        inicio
            escribir("Introduce el primero operando (A)")
            leer(a)
            escribir("Introduce el operador (+, -, *, /)")
            leer(op)
            escribir("Introduce el segundo operando (B)")
            leer(b)

            según (op) hacer
                inicio
                    '+': r = a + b
                        escribir("Resultado: ", r)
                    '-': r = a - b
                        escribir("Resultado: ", r)
                    '*': r = a * b

```

```

        escribir("Resultado: ", r)
    '/': r = a / b
        escribir("Resultado: ", r)
    si_no: escribir("Error: la operación ", op, " es desconocida")

fin    // fin-según

    escribir("¿Desea terminar (S/N)?")
    leer(terminar)
fin
mientras (terminar == 'N')
fin

```

48. Juego del número secreto. El ordenador elegirá un número al azar entre 1 y 100. El usuario irá introduciendo números por teclado, y el ordenador le irá dando pistas: "mi número es mayor" o "mi número es menor", hasta que el usuario acierte. Entonces el ordenador le felicitará y le comunicará el número de intentos que necesitó para acertar el número secreto.

```

algoritmo juego_número_secreto
variables
    num_secreto, n, intentos son enteros
inicio
    num_secreto = aleatorio(100)        // Aleatorio entre 0 y 99
    num_secreto = num_secreto + 1        // Aleatorio entre 1 y 100
    intentos = 0

    repetir
    inicio
        escribir("Intenta adivinar mi número secreto")
        leer(n)
        intentos = intentos + 1

        si (num_secreto < n)
            escribir("Mi número secreto es MENOR")
        si (num_secreto > n)
            escribir("Mi número secreto es MAYOR")
        si (num_secreto == n)
            escribir("ENHORABUENA, has acertado el número en ", intentos, " intentos")
    fin
    mientras (num_secreto != n)
fin

```