1. Calcular **el área y la circunferencia** de un círculo cuyo radio será proporcionado a través del teclado. Recuerda que área = πr^2 y circunferencia = $2 \pi r$
2. Determinar si un número leído del teclado es **positivo o negativo**.
3. Calcular la **raíz cuadrada** de un número introducido por teclado. Hay que tener la precaución de comprobar que el número sea positivo.
4. Leídos dos números por teclado, A y B, calcular la resta del **mayor menos el menor**. Por ejemplo, si A = 8 y B = 3, el resultado debe ser A – B, es decir, 5. Pero si A = 4 y B = 7, el resultado debe ser B – A, es decir, 3.
5. Determinar si un año es **bisiesto** o no (los años bisiestos son múltiplos de 4; utilícese el operador módulo)
6. Averiguar si un número real introducido por teclado tiene o no **parte fraccionaria** (utilícese la función trunc() que describimos en el tema 1)
7. 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.
8. Leer tres números por teclado, X, Y y Z, y decidir **si están ordenados** de menor a mayor.
9. Como el anterior, pero para averiguar si los números son **consecutivos**.
10. 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.
11. 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.
12. Dados tres números enteros, A, B, C, determinar cuál es el **mayor**, cuál el **menor** y cuál el **mediano**. Sólo pseudocódigo.
13. Escribir todos los **números pares** entre **1 y 1000**
14. 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.
15. Calcular la **suma de todos los números pares** entre 1 y 1000. Es decir, 2 + 4 + 6 + ... + 998 + 1000.
16. Mostrar el mensaje **"¿Desea terminar? (S/N)"** y leer la respuesta del usuario. Si es "S", el programa terminará. Si es "N", volverá a formular la pregunta.
17. 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.
18. 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.
19. Calcular el valor **máximo** de una serie de 10 números introducidos por teclado. Sólo en pseudocódigo.
20. Generalizar el ejercicio anterior para que también se averigüe el valor **mínimo** y el **medio**. Sólo en pseudocódigo.
21. 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 x 4 x 3 x 2 x 1. Sólo en pseudocódigo.
22. 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. Sólo en pseudocódigo.
23. Generalizar el algoritmo anterior para averiguar **todos los números primos** que existen entre 2 y 1000. Sólo en pseudocódigo.
24. 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). Sólo pseudocódigo.
25. Generar combinaciones al azar para la **lotería primitiva** (6 números entre 1 y 49). Debes utilizar la función *aleatorio(x)* que vimos en el tema 1. Por ahora, no te preocupes porque los números puedan repetirse.
26. Generar combinaciones al azar para la **quiniela** (14 valores dentro del conjunto 1, X o 2)
27. 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)
28. **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. Sólo en pseudocódigo.

29. El siguiente algoritmo intenta calcular la **tabla de multiplicar** de un número N introducido por teclado, pero tiene dos errores que debes corregir.

```
algoritmo tabla_multiplicar
variables
  N es entero
inicio
  leer(N)
  cont = 1
  mientras (cont <= 10) hacer
  inicio
    escribir(N*cont)
  fin
fin
```

30. El siguiente algoritmo lee dos números por teclado, A y B, y determina si **B es divisor de A**, pero cuando B = 0 se produce un error de intento de división entre 0 en la instrucción A mod B. Corrígelo para que no ocurra:

```
algoritmo divisor
variables
  A,B son enteros
inicio
  leer(A,B)
  si (A mod B == 0) entonces
    escribir('B es divisor de A')
  si_no
    escribir('A es divisor de B')
fin
```

31. El siguiente algoritmo sirve para **contar hacia atrás** desde un número N hasta 1, pero tiene algunos fallos. Corrígelos:

```
algoritmo contar_hacia_atrás
variables
  N, cont son enteros
inicio
  leer(N)
  repetir
  inicio
    cont = cont - 1
    escribir(cont)
  fin
  mientras que (cont <= 1)
fin
```

32. Averigua **qué hace** este algoritmo:

```
algoritmo misterioso
variables
  A, limite, cont son enteros
inicio
  leer (A)
  leer (límite)
  para cont desde A hasta límite hacer
  inicio
    escribir (A*cont)
  fin
fin
```

33. Averigua **qué hace** este algoritmo:

```
algoritmo misterioso_2
variables
  A, B, C son enteros
inicio
  leer(A,B,C)
  si (A > B) y (A > C) entonces
  inicio
    si (B > C) entonces
    inicio
      escribir (A-B-C)
    fin
  fin
  si_no
  inicio
    si (A < B) y (A < C) entonces
    inicio
      si (B < C) entonces
      inicio
        escribir (A+B+C)
      fin
    fin
  inicio
fin
```

34. Averigua **qué hace** este algoritmo.

```
algoritmo misterioso_3
variables
  nombre es cadena
  dinero es entero
  valor1, valor2, valor3 son enteros
  continuar es carácter
inicio
  escribir('Introduzca su nombre')
  leer(nombre)
  escribir('Bienvenido al juego, ', nombre)
  dinero = 50
  repetir
  inicio
    valor1 = aleatorio(9)
    valor2 = aleatorio(9)
    valor3 = aleatorio(9)
    escribir('Su jugada es:')
    escribir(valor1, valor2, valor3)
    si (valor1 == valor2) y (valor1 == valor3) entonces
    inicio
      dinero = dinero + 100
      escribir ('¡Enhorabuena, ', nombre, '! Ha ganado 100€')
    fin
    si_no
    inicio
      si (valor1 == valor2) o (valor1 == valor3) o (valor2 == valor3) entonces
        dinero = dinero + 20
      si_no
        dinero = dinero - 10
    fin
    escribir('Su saldo actual es de ', dinero, ' euros')
    escribir('¿Desea seguir jugando? (S/N)')
    leer(continuar)
  fin
  mientras que (continuar == 'S')
  escribir('¡Hasta la próxima!)
fin
```

35. Introduce las siguientes **modificaciones** en el algoritmo anterior:

a. *Al finalizar el juego, imprimir un texto con la cantidad de dinero que le queda al jugador*
b. *Si en algún momento el jugador se queda sin saldo, el juego debe terminar automáticamente, sin siquiera preguntar si desea continuar.*
c. *Introducir un nivel de dificultad seleccionable por el jugador, de manera que la probabilidad de acertar sea proporcional a ese nivel de dificultad (modificando las funciones aleatorias).*