

Relación de Problemas (Tema 1)

1. Escribe un programa que lea desde el teclado dos valores de tipo INTEGER y escriba en pantalla su suma, su diferencia, su producto, su cociente y su resto.
2. Escribe un programa que pida una letra desde el teclado, la pase a mayúscula e imprima el resultado por pantalla.
3. Escribe un programa que lea el radio y la altura de un cilindro desde el teclado y escriba por pantalla el área y volumen correspondiente.

NOTA: El área y volumen del cilindro de altura h y radio R es:

$$\text{Area} = 2\pi R^2 + 2\pi Rh$$

$$\text{Volumen} = \pi R^2 h$$

4. Escribe un programa que lea desde el teclado una cantidad de segundos y muestre por pantalla el número de horas, minutos y segundos equivalentes.
5. Escribe un programa que lea un carácter del teclado y escriba por pantalla su código dentro de la tabla ASCII.
6. a) ¿Qué ocurre al ejecutar el siguiente programa?

```
PROGRAM Prueba;  
  
VAR  
  
  Num : INTEGER;  
  
BEGIN  
  
  Num := HIGH(INTEGER);  
  
  Num := Num + 10 - 100;  
  
  Writeln (Num)  
  
END.
```

b) ¿Cómo puede solucionarse el problema?

7. Escribe un programa que lea del teclado la longitud de los dos catetos de un triángulo rectángulo, calcule la longitud de la hipotenusa y la escriba por pantalla.
8. Escribe un programa que lea la longitud del lado de un cuadrado y escriba en pantalla su área y su perímetro.
9. Escribe un programa que lea un carácter del teclado y escriba en pantalla el carácter que le sigue en el código ASCII.

Relación de Problemas (Tema 2)

1. Si la variable `valor` es de tipo `BOOLEAN`, ¿cómo se puede escribir de otro modo la condición o pregunta `valor = TRUE?`, ¿y `valor = FALSE?`.
2. Si la variable `cierto` es de tipo `BOOLEAN`, y `a`, `b` son de tipo `INTEGER`, ¿a qué son equivalentes las siguientes expresiones ?

i) `cierto := (a < b) AND (b < a)`

ii) `cierto := (a <= b) OR (b <= a)`

3. Si `x` e `y` son variables de tipo `BOOLEAN`, ¿cuál es la relación entre las siguientes dos expresiones?:

i) `x <> y`

ii) `(x OR y) AND NOT (x AND y)`

4. Si `p`, `q` y `r` son variables booleanas, comprueba las siguientes igualdades:

i) `(p OR r) AND (q OR r) ≡ (p AND q) OR r`

ii) `(p AND r) OR (q AND r) ≡ (p OR q) AND r`

5. Escribe un programa que determine el mayor de tres números enteros.
6. Escribe un programa que sin utilizar sentencias de asignación escriba ordenados de menor a mayor tres números almacenados en variables de tipo entero.
7. Escribe un programa que dado un número entero introducido por teclado compruebe si es par o no.
8. Escribe un programa que lea tres números `d`, `m` y `a` que representan una fecha (día, mes y año) y calcule el día de la semana correspondiente usando la congruencia de Zeller:

$$\text{día} = (700 + (26x - 2) \text{DIV } 10 + d + y + y \text{DIV } 4 + z \text{DIV } 4 - 2z) \text{MOD } 7$$

donde los valores de `x`, `y`, `z` son:

	si $m \leq 2$	si $m \geq 3$
x	$m + 10$	$m - 2$
y	$(a - 1) \text{MOD } 100$	$a \text{MOD } 100$
z	$(a - 1) \text{DIV } 100$	$a \text{DIV } 100$

El valor de día estará entre 0 y 6, (0 significa Domingo, 1 Lunes, etc.)

NOTA: Este algoritmo es solo válido para el calendario gregoriano, que fue introducido en distintos países en distintas fechas (el 14 de Septiembre de 1942 en Inglaterra, por ejemplo).

9. Escribe un programa que determine el menor número de monedas y billetes de curso legal equivalentes a cierta cantidad de pesetas leídas del teclado.

NOTA: Considera que existen billetes de 10.000, 5.000, 2.000, 1.000 y monedas de 500, 100, 25, 5 y 1 pesetas.

10. Escribe un programa que lea dos letras minúsculas y determine:

- el número de letras que las separan
 - la letra o dos letras centrales del intervalo que determinan, pero en mayúsculas
11. Escribe un programa que determine si un año es bisiesto. Un año es bisiesto si es múltiplo de 4 (por ejemplo 1984). Sin embargo, los años múltiplos de 100 sólo son bisiestos cuando a su vez son múltiplos de 400 (por ejemplo 1800 no es bisiesto, mientras que 2000 lo será).
 12. Escribe un programa que calcule el número de días de un mes, dados los valores numéricos del mes y año.

NOTA: considera los años bisiestos para Febrero.

13. El domingo de Pascua es el primer domingo después de la primera luna llena posterior al equinoccio de primavera, y se determina mediante el siguiente cálculo sencillo:

$$A = \text{año MOD } 19$$

$$B = \text{año MOD } 4$$

$$C = \text{año MOD } 7$$

$$D = (19 * A + 24) \text{ MOD } 30$$

$$E = (2 * B + 4 * C + 6 * D + 5) \text{ MOD } 7$$

$$N = (22 + D + E)$$

Donde N indica el día del mes de Marzo si N es menor o igual que 31 o Abril si es mayor que 31 (el 32 indicaría el 1 de Abril). Construye un programa que determine fechas de domingos de Pascua.

14. Escribe un programa que dados los conjuntos de puntos pertenecientes a la circunferencia cuya ecuación es $x^2 + y^2 = 16$, a la elipse $x^2/36 + y^2/16 = 1$ y a la recta $y = 2x + 1$, indique para una pareja de coordenadas x e y , leídas desde el teclado, el conjunto o conjuntos a los que pertenece.
15. Un sistema de ecuaciones lineales:

$$ax + by = c$$

$$dx + ey = f$$

se puede resolver con las siguientes fórmulas:

$$x = (ce - bf) / (ae - bd)$$

$$y = (af - cd) / (ae - bd)$$

Diseña el programa que lea los coeficientes a, b, c, d, e, f y muestre los valores de x e y .

NOTA: Estudia el caso en que $(ae - bd) = 0$.

16. Desarrolla un programa que calcule las raíces correspondientes a una ecuación de segundo grado $ax^2 + bx + c = 0$.
- NOTA:** Estudia el caso de raíces complejas (valor del discriminante < 0) y el caso $a = 0$.
17. Escribe un programa que lea un número de tipo **CARDINAL**, y si tiene exactamente cuatro cifras, escriba si es capicúa o no.
18. Escribe un programa que lea un número de tipo **CARDINAL**, y si tiene exactamente cuatro cifras, escriba si el número es igual a la suma de los cuadrados de sus cifras.

Relación de problemas (Tema 3)

1. Escribe un programa que lea del teclado una lista de valores reales positivos terminada por un número negativo y calcule la media de esta lista.
2. Escribe un programa que lea un número entero positivo por teclado y determine si es primo o compuesto.
3. Diseña un programa que escriba en pantalla la tabla de multiplicar de un número del 1 al 9 introducido desde el teclado.
4. Escribe un programa que lea del teclado una lista de valores enteros positivos terminada por 0 y determine los valores máximo y mínimo.
5. Escribe un programa que calcule e imprima en pantalla los N primeros números primos, siendo N un número que se introduce por teclado.
6. Escribe un programa que calcule el valor de seno x dado por:

$$\text{Seno } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Este programa debe leer dos variables `x` y `precisión` donde `x` es un ángulo en radianes y `precisión` debe ser positiva y menor que 1. La serie debe sumarse hasta que el valor absoluto del último sumando sea menor que `precisión`, incluyendo este último sumando en la suma.

NOTA: El término con numerador x^n de la serie se puede calcular multiplicando el previo

por $-\frac{x^2}{n \times (n-1)}$

7. Si x_0 es una aproximación a \sqrt{a} entonces una aproximación mejor es:

$$x_1 = \frac{1}{2} \left(x_0 + \frac{a}{x_0} \right)$$

Podemos repetir esto con x_1 y obtener una nueva aproximación. Este proceso, que es una aplicación del método de Newton-Raphson, se puede generalizar a:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right), n = 1, 2, \dots$$

Escribe un programa que lea los valores de a y *precisión*. El programa debe calcular la raíz de a utilizando el método descrito. El valor inicial para la iteración será $a/2$. El programa debe mostrar en pantalla todas las aproximaciones obtenidas.

8. Escribe tres programas para calcular el factorial de un número entero leído desde el teclado utilizando las sentencias FOR, WHILE y REPEAT.
9. Escribe un programa que visualice en pantalla una figura similar a la siguiente:

```
*
**
***
****
```

Siendo el número de líneas un dato que se lee al principio del programa.

10. Un número es perfecto si coincide con la suma de todos sus divisores excepto el mismo. Por ejemplo 28 es perfecto ya que sus divisores son 1, 2, 4, 7, 14 y $1+2+4+7+14 = 28$. Escribe un programa que determine si un número leído del teclado es perfecto.
11. Para encontrar el máximo común divisor (mcd) de dos números enteros se puede emplear el algoritmo de Euclides: dados los números a y b (siendo $a > b$) se divide a por b obteniendo el cociente q_1 y el resto r_1 . Si $r_1 \neq 0$, se divide b por r_1 , obteniendo el cociente q_2 , y el resto r_2 . Si $r_2 \neq 0$ se divide r_1 por r_2 obteniendo cocientes y restos sucesivos. El proceso continúa hasta obtener un resto igual a 0. El resto anterior a éste es el máximo común divisor de los números iniciales (a y b). Escribe un programa que lea dos números enteros positivos del teclado y calcule su máximo común divisor y su mínimo común múltiplo (mcm).

NOTA: obtener el mínimo común múltiplo a partir de la siguiente igualdad:

$$\text{mcd}(a,b) * \text{mcm}(a,b) = a * b$$

12. Diseña un programa que calcule el cociente entero y el resto correspondiente para dos números naturales dados por teclado, sin utilizar los operadores DIV y MOD.
13. Escribe un programa que lea un número N y calcule el término N -ésimo de la sucesión de Fibonacci:

$$\text{Fibonacci}(1) = 0$$

$$\text{Fibonacci}(2) = 1$$

$$\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2), \text{ si } n > 2$$

14. Escribe un programa que calcule y visualice el más grande, el más pequeño y la media de N números. El valor N se solicitará al principio del programa y los números serán introducidos por teclado.
15. Escribe un programa para encontrar el número natural N más pequeño tal que la suma de los N primeros números enteros exceda una cantidad introducida por teclado.
16. El valor $\exp(x)$ se puede calcular mediante la siguiente serie:

$$\exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Escribe un programa que lea un valor x de teclado y calcule el valor de $\exp(x)$ deteniendo el cálculo cuando el valor del último término sumado sea inferior a una diezmilésima.

17. Escribe un programa que calcule una aproximación al número π sabiendo que

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

para ello sumará términos de la serie anterior mientras que sus valores absolutos sean mayores a una diezmilésima y del valor obtenido despejará π .

18. Diseña un programa que multiplique dos números enteros mediante el algoritmo de la multiplicación rusa. Este algoritmo multiplica por 2 el multiplicando y divide por dos el multiplicador hasta que el multiplicador toma el valor 1. Después suma todos los multiplicandos correspondientes a multiplicadores impares. Dicha suma es el producto de los números originales.

Ejemplo: $37 * 12 = 444$ (multiplicador: 37, multiplicando:12)

```

37  12
18  24
 9  48
 4  96
 2  192
 1  384

```

$12 + 48 + 384 = 444$

19. Escribe un algoritmo que lea un número natural N y un carácter. La salida debe ser un rombo compuesto del carácter y de la anchura que especifica el número N . Por ejemplo, si

N es 5 y el carácter es *, el rombo sería:

```
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * *
  * * * *
   * * *
    * *
     *
      *
```

20. Escribe un algoritmo que imprima una pirámide de dígitos como la de la figura, tomando como entrada el número de filas de la misma siendo este número menor o igual que 9.

```
      1
     1 2 1
    1 2 3 2 1
   1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1
```

Relación de problemas (Tema 4)

1. Supongamos que tenemos un programa principal con las siguientes variables:

```

VAR
    x, y : REAL;
    m    : INTEGER;
    c    : CHAR;
    n    : CARDINAL;
  
```

y la siguiente declaración de subprograma:

```

PROCEDURE Prueba (a, b : INTEGER; VAR c, d : REAL;
                 VAR e : CHAR);
  
```

Averigua cuáles de las siguientes llamadas son válidas:

- 1) Prueba (m+3,10,x,y,c);
 - 2) Prueba (n+3,10,x,y,c);
 - 3) Prueba (m, 19, x, y);
 - 4) Prueba (m, m*m, y, x, c);
 - 5) Prueba (m, 10, 35.0, y, 'E');
 - 6) Prueba (30, 10, x, x+y, c);
2. Escribe una función que calcule las combinaciones de m elementos tomados de n en n . Para ello escribe primero otra función que calcule el factorial de un número.

NOTA: Recuérdese que

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

Escribe otra función más eficiente que no necesite calcular el factorial.

3. Escribe una función que tome tres parámetros, dos de tipo REAL y uno de tipo CHAR. La función debe sumar, restar, multiplicar o dividir los valores de los dos primeros parámetros dependiendo del código indicado en el tercero (+,-,*,/) y devolver el resultado. Al diseñar esta función, supón que el tercer argumento es necesariamente uno de los cuatro (+,-,*,/).

Escribe un programa que ofrezca al usuario la posibilidad de realizar una de esas cuatro operaciones o salir del programa hasta que el usuario decida abandonar el programa, lea los operandos necesarios para realizar la operación, utilice la función anterior para hacer los cálculos y muestre el resultado en pantalla. Asegúrate de que el programa utiliza la función anterior de forma correcta.

4. Dos números a y b se dice que son amigos si la suma de los divisores de a (salvo el mismo) coincide con b y viceversa. Diseña un programa que tenga como entrada dos números naturales n y m , y muestre por pantalla todas las parejas de números amigos que existan en el intervalo determinado por n y m .

5. Considera el siguiente procedimiento:

```
PROCEDURE Escr (C : CHAR; Long : INTEGER);
BEGIN
  WHILE Long > 0 DO
    BEGIN
      Write (C);
      Long := Long - 1
    END
  END;
END;
```

a) Si Ch tiene el valor 'X' y Numero el valor 5, ¿cuál sería el efecto de ejecutar cada una de las siguientes llamadas al procedimiento?:

```
Escr(Ch,4*Numero-12)      Escr(Ch,6)      Escr(5,Numero)
Escr('/',Numero)          Escr('.',6)      Escr('p',-10)
```

b) Escribe llamadas al procedimiento `Escr` para que cuando se ejecuten produzcan las siguientes salidas:

- 35 guiones sucesivos
- 6 veces tantos espacios en blanco como el valor de `Numero`
- el valor actual de `ch` 14 veces

6. Dadas las siguientes declaraciones en un determinado algoritmo:

```
VAR a,b,c : CARDINAL;
    si : BOOLEAN;
...
FUNCTION Uno(x, y : CARDINAL) : BOOLEAN;
...
PROCEDURE Dos(VAR x : CARDINAL; y : CARDINAL);
...
FUNCTION Tres(x : CARDINAL) : CARDINAL;
```

¿ Cuáles de las siguientes llamadas a subprograma son válidas?

- a) `IF Uno(a,b) THEN...`
- b) `Dos(a,b+3)`
- c) `si := Uno(c,5)`
- d) `si := Dos(c,5)`
- e) `Dos(a,Tres(a))`

- f) Dos(Tres(b),c)
- g) **IF** Tres(a) **THEN** ...
- h) b := Tres(Dos(a,5))
- i) Dos(4,c)

7. Escribe un algoritmo que tome como entrada desde teclado dos números naturales N e i , e imprima en pantalla el dígito que ocupa la posición i -ésima del número N . Si i es mayor que el número de dígitos de N , se escribirá en pantalla el valor -1. Por ejemplo, para $N = 25064$ e $i = 2$, el resultado es el dígito 6, y para $i = 7$, el resultado es -1.
8. ¿Qué salida produce por pantalla la ejecución del siguiente algoritmo?

```
PROGRAM Anidado;
```

```
VAR
```

```
  a, b, c, x, y : CARDINAL;
```

```
PROCEDURE Primero;
```

```
  BEGIN
```

```
    a := 3 * a;
```

```
    c := c + 4;
```

```
    Write ('Primero');
```

```
    Writeln (a,b,c)
```

```
  END;
```

```
PROCEDURE Segundo;
```

```
  VAR b : CARDINAL;
```

```
  BEGIN
```

```
    b := 8;
```

```
    c := a + c + b DIV 3;
```

```
    Write ('Segundo');
```

```
    Writeln (a, b, c)
```

```
  END;
```

```
PROCEDURE Tercero (VAR x : CARDINAL; y : CARDINAL);
```

```
  BEGIN
```

```
    x := x + 4;
```

```
    y := y + 1;
```

```
    Write ('Tercero');
```

```

        Writeln (a, b, c, x)

    END;

BEGIN

    a := 3; b := 2;

    c := 1; x := 11;

    y := 22;

    Primero; Segundo;

    Tercero (a,b);

    Write ('Anidado');

    Write (a, b, c, x, y)

END.

```

9. Realiza un procedimiento que intercambie el valor de dos variables de tipo CHAR.
10. Escribe un procedimiento que, dadas las coordenadas polares de un número complejo (r , q), obtenga las correspondientes cartesianas (x , y).

NOTA: Recuerda que

$$x = r \cdot \cos q$$

$$y = r \cdot \sen q$$

11. Escribe un programa que lea un número positivo del teclado y escriba por pantalla los números perfectos menores a él. Para ello define primero una función `EsPerfecto` que tome como parámetro un número y devuelva `TRUE` o `FALSE` dependiendo de que sea perfecto o no.
12. Escribe un algoritmo que acepte como entrada desde teclado un número entero positivo. El algoritmo debe producir como salida el resultado de sumar dos a dos los dígitos que aparecen en posiciones simétricas respecto al dígito central dentro del número dado como entrada. Por ejemplo, para el número 23548 la salida es: $2+8 = 10$, $3 + 4 = 7$, 5 para el número 6582 la salida es: $6 + 2 = 8$, $5 + 8 = 13$
13. Escribe un subprograma para calcular el máximo común divisor de cuatro números utilizando para ello otro subprograma que calcule el máximo común divisor de dos números (algoritmo de Euclides).
14. Escribe un programa que lea dos enteros positivos correspondientes a un año y un mes, y escriba el correspondiente calendario de la siguiente forma:

```

    D L M X J V S
                1
    2 3 4 5 6 7 8
    9 10 11 12 13 14 15
    16 17 18 19 20 21 22

```

23 24 25 26 27 28 29

Este problema puede dividirse en las siguientes partes:

- 1) Leer los valores del mes y año
- 2) Encontrar el día de la semana en que empieza el mes (ver ejercicio sobre congruencia de Zeller).
- 3) Averiguar cuántos días tiene el mes.
- 4) Escribir el correspondiente calendario.

Usa un subprograma separado para cada parte.

15. Escribe un programa que lea un número positivo del teclado y escriba por pantalla los números primos menores a él. Para ello define primero una función `EsPrimo` que tome como parámetro un número y devuelva `TRUE` o `FALSE` dependiendo de que sea primo o no.
16. Escribe un programa que calcule el máximo de tres números enteros, definiendo previamente una función que calcule el máximo de dos.

Relación de Problemas (Tema 5)

1. Escribe un programa que solucione el sistema de ecuaciones:

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Para ello diseña un subprograma que tome como parámetros por valor a , b , c , d , e y f , como parámetros de salida por referencia x , y y Estado, donde Estado es del tipo enumerado:

TYPE Estado = (Determinado, Indeterminado, Incompatible);

Si el sistema está determinado x e y contendrán la solución al sistema y Estado valdrá Determinado. En otro caso Estado valdrá Indeterminado o Incompatible y se enviará un mensaje al usuario indicando el estado del sistema.

El programa principal deberá leer los coeficientes, llamar al subprograma y escribir el resultado por pantalla.

NOTA: Si $ae - bd = 0$ y $bf - ec = 0$ el sistema es indeterminado. Si $ae - bd = 0$ pero $bf - ec \neq 0$ el sistema es incompatible.

2. Escribe una función que tome como parámetros dos funciones f y g ambas de rango y dominio REAL, un valor x de tipo REAL y devuelva f compuesta de g evaluada en el punto x . Prueba la función con casos concretos de f y g .

NOTA: $(f \circ g)(x) = f(g(x))$

3. Escribe un subprograma que tome como parámetro una función f de rango y dominio REAL y un valor x de tipo REAL, y devuelva un valor aproximado de la derivada de f en dicho punto utilizando para ello la siguiente fórmula con ϵ igual a una millonésima.

$$f'(x) = \frac{f(x + \epsilon) - f(x)}{\epsilon}$$

Escribe un programa que utilice la función anterior para calcular el valor de la derivada de las funciones seno y coseno en los puntos π y $\pi/2$.

4. Escribe una función que encuentre un cero de una función utilizando el algoritmo de bipartición. Este algoritmo toma como parámetros, entre otros, una función f (de dominio y rango REAL que suponemos continua), y tres valores a , b y Epsilon de tipo REAL, y devuelve el cero calculado. El algoritmo debe comprobar que el valor de b sea mayor que a y que el signo de $f(a)$ es distinto que el signo de $f(b)$ (condiciones del Teorema de Bolzano). A partir de aquí se repite el siguiente proceso: calcular el punto medio del intervalo $[a,b]$ y llamarlo c . Si $f(c) = 0$, entonces hemos encontrado un cero de f en c y acabamos. En otro caso determinamos en qué mitad del intervalo $[a,b]$ se produce el cambio de signo ($[a,c]$ o $[c,b]$) y repetimos el proceso con dicha mitad. Todo este proceso se repite hasta que se da la condición anterior ($f(c) = 0$) o hasta que el ancho del intervalo considerado sea menor que Epsilon. Utiliza la función anterior para encontrar un cero de la función coseno en $[0, \pi]$.
5. Escribe una función Pliega que tiene cuatro parámetros: f , a , b , Inicial. El primer parámetro es una función que toma dos enteros y devuelve un entero. Los otros tres parámetros son valores enteros. El resultado de la función Pliega es de tipo entero y debe ser el resultado de calcular la expresión:

$$f(f(\dots f(f(f(\text{Inicial}, a), a+1), a+2), \dots b-1), b).$$

Es decir, calculamos primero f con parámetros Inicial y a y obtenemos un resultado. Con este resultado y $a+1$ volvemos a calcular f . Seguimos así hasta que llegamos a que el segundo parámetro es b .

Escribe un programa que utilice la función `Pliega` para calcular el factorial de un número N leído de teclado y el sumatorio de los primeros N naturales, donde N es un número leído del teclado

6. Realiza un procedimiento `Filtra` que toma tres parámetros: f , a y b . f es una función que toma un entero y devuelve un booleano. a y b son dos valores enteros. `Filtra` deberá escribir por pantalla todos los valores enteros x en el intervalo $[a..b]$ para los cuales $f(x)$ sea cierto.

Utiliza el procedimiento `Filtra` en un programa que escriba los números perfectos y los números primos menores a un valor `Maximo` dado por teclado.

7. Escribir una función que calcule el valor de la integral definida

$$\int_a^b f(x)dx.$$

Esta función tomará como parámetros: una función f de dominio y rango `REAL`, los valores a y b de tipo `REAL`, y un parámetro `Epsilon` también `REAL`. Para ello dividiremos el intervalo $[a,b]$ en intervalos de amplitud menor o igual a `Epsilon` y aproximaremos la integral de cada subintervalo con la fórmula del trapecio:

$$\int_{x_0}^{x_1} f(x)dx \cong \frac{x_1 - x_0}{2} [f(x_0) + f(x_1)]$$

El valor de $\int_a^b f(x)dx$ será el sumatorio de las aproximaciones anteriores. Utilizar la función anterior para calcular la integral definida de la función seno entre 0 y $\pi/2$.

Relación de problemas (Tema 6)

1. Escribe un programa que permita leer dos matrices de tamaño 10x10 de números reales del teclado, calcular la matriz producto y escribirla por pantalla.
2. Escribe un programa que permita leer una matriz de tamaño 5x6 de números reales del teclado, calcular la matriz traspuesta y escribirla por pantalla.
3. Escribe un programa que convierta un número entero positivo expresado en cualquier base natural entre 2 y 10 a una base natural entre 2 y 10. Para ello se leerá del teclado la base inicial, el número y la base destino.
4. Dadas las siguientes declaraciones:

TYPE

```

NOMBRE                = ARRAY [0..25] OF CHAR;

REGISTROLISTA         = RECORD
                        Lista      : ARRAY [1..5000] OF NOMBRE;
                        Sublista   : ARRAY [1..20] OF NOMBRE;
END;

REGISTROCATALOGO = RECORD
                        NombreElemento : NOMBRE;
                        TipoElemento   : (Uno, Dos);
                        Subcatalogo     : REGISTROLISTA;
END;

```

VAR

```

Catalogo      : ARRAY [1..20] OF REGISTROCATALOGO;
UnCatalogo    : REGISTROCATALOGO;
UnaLista      : REGISTROLISTA;
UnNombre      : NOMBRE;

```

¿Cuáles de las siguientes sentencias son válidas ?

- 1) **IF** (REGISTROCATALOGO.TipoElemento = Uno) **THEN** ...
- 2) Catalogo[1].Subcatalogo.Lista[2] := UnCatalogo;
- 3) Catalogo[5].RegistroLista := UnaLista;
- 4) UnCatalogo.NombreElemento[2] := UnaLista.Lista[2];
- 5) Catalogo[1].Lista.Lista[1,2] := UnNombre[2];
- 6) Catalogo[2].Lista[1] := UnNombre;
- 7) **IF** Catalogo[20].TipoElemento = Dos **THEN**
Catalogo[10].RegistroLista.Sublista[3,9] := 'A'

```
8) UnCatalogo := Catalogo[5];
```

```
9) UnCatalogo.Subcatalogo := UnaLista;
```

5. Escribe un programa que sume dos vectores de tipo base INTEGER e imprima el resultado en pantalla. Se consideran datos de entrada la dimensión de los vectores y las componentes de cada uno. Representa cada vector como un registro de dos componentes (el vector propiamente dicho y la dimensión utilizada). Declara en el programa tres subprogramas: lectura de un vector desde teclado, suma de dos vectores y escritura de un vector por pantalla. El programa debe funcionar para vectores de como máximo 10 componentes.

6. Escribe un programa que permita leer dos matrices de números reales del teclado, calcular la matriz producto y escribirla por pantalla. El programa debe ser capaz de operar con matrices de tamaño menor o igual a 30x30. El tipo que represente la matriz debe ser un registro con tres componentes: la matriz, el número de filas y el número de columnas útiles. Escribe subprogramas para cada una de las tareas.

Nota: Detecta posibles errores en las dimensiones de las matrices.

7. Escribir un subprograma que tome como parámetro una matriz cuadrada y devuelva TRUE si es simétrica o FALSE en otro caso. Una matriz A es simétrica si $\forall i, j \quad A_{ij} = A_{ji}$

Nota: Detecta posibles errores en la dimensión de las matriz.

8. Escribe un programa que lea un texto del teclado y escriba la frecuencia de aparición de cada vocal en él (cuidado con las vocales acentuadas).

9. Escribe un programa que lea una palabra del teclado y determine si dicha palabra es un identificador válido en Pascal.

10. Se desea efectuar operaciones de suma de cantidades enteras que se caracterizan por tener una gran cantidad de dígitos, de manera que queda excluido el uso del tipo REAL. Para ello cada número se representará como una cadena de caracteres. Escribe un subprograma que realice la operación anterior.

Nota: Se considera que no se va a trabajar con números de más de 100 dígitos.

11. Escribe un programa que simule el comportamiento de una calculadora simple para aritmética fraccional expresando el resultado en forma fraccional y simplificado al máximo el resultado. Para ello se definirá el tipo RACIONAL como un registro de dos componentes enteras (Numerador y Denominador), y subprogramas para leer del teclado, escribir por pantalla, simplificar, sumar, restar, multiplicar, dividir y comparar números racionales.

Nota: El subprograma que simplifica un número racional debe dividir el numerador y el denominador por el máximo común divisor de ambos, y en caso de que el número racional sea negativo dejar este signo en el numerador.

12. Escribe un programa que lea dos puntos bidimensionales representados como registros y calcule la longitud del segmento que los une y la pendiente de la recta que pasa por ellos.

Nota: recuerda que una recta viene determinada por dos puntos (x_1, y_1) , (x_2, y_2) siempre que sean distintos. Su pendiente en función de estos dos puntos es $(y_2 - y_1) / (x_2 - x_1)$.

13. Supongamos que deseamos evaluar a un determinado número de alumnos siguiendo el criterio de que aprobará aquel que supere la nota media de la clase.

Escribe un programa que lea un número indeterminado de alumnos (como máximo 20), y las notas de tres evaluaciones, y como resultado emita un informe indicando para cada alumno

las evaluaciones que tiene aprobadas y suspensas.

Ejemplo de la salida que se debe obtener.

Alumno	Nota-1	Nota-2	Nota-3
Juan López	Aprobado	Suspense	Aprobado
Luis García	Suspense	Aprobado	Aprobado
Pedro Ruiz	Aprobado	Aprobado	Aprobado

14. Definir un tipo registro variante (*FIGURA*) de modo que una variable de dicho tipo pueda almacenar un círculo, o un triángulo rectángulo o un rectángulo o bien un cuadrado. Según sea la figura se desean almacenar los siguientes datos:

- Si es un círculo, el radio.
- Si es un triángulo, la base y la altura.
- Si es un rectángulo, la base y la altura.
- Si es un cuadrado, el lado.

Escribir funciones que tomen como parámetro una *FIGURA* y calculen el área y perímetro de éstas.

15. El algoritmo de *La Criba de Eratóstenes* puede ser usado para calcular los números primos menores o iguales a un *N* dado:

- 1) Crear un conjunto llamado *Criba* que contenga todos los números naturales en el intervalo $[2, N]$.
- 2) Crear un conjunto vacío y llamarlo *Primos*.
- 3) Repetir los siguientes pasos
 - 4) Tomar el primer número que quede en *Criba*.
 - 5) Este número será primo, por lo que se añade al conjunto *Primos*.
 - 6) Quitar este número y todos sus múltiplos del conjunto *Criba*.
- 7) Hasta que *Criba* quede vacía.
- 8) El conjunto *Primos* contiene todos los primos en el intervalo $[2, N]$.

Utilizar el algoritmo anterior para escribir un programa que calcule y muestre los primos menores o iguales a 100.

16. Escribe un programa que se comporte como una calculadora que pida repetidamente un operador de conjuntos y dos operandos que sean conjuntos de letras minúsculas y que escriba el resultado de la operación. Las operaciones se expresan como caracteres, siendo válidas las siguientes:

+	Unión de conjuntos
-	Diferencia de conjuntos
*	Intersección de conjuntos

El proceso se repetirá hasta que se introduzca como código de operación el carácter '&'. Los operadores y el resultado se expresan como cadenas de caracteres.

Ejemplo:

Operación = *

Operando1 = azufre
 Operando2 = zafio
 Resultado = afz
 Operación = -
 Operando1 = abracadabra
 Operando2 = abaco
 Resultado = rd
 Operación = &
 FIN

17. Disponemos de un vector de 20 números enteros. Diseña un programa que mantenga en dicho vector los números pares separados de los impares, de forma que cada uno de los subconjuntos estén ordenados entre sí de forma ascendente. Diseña otro subprograma que ordene un vector como el anterior mediante intercalación de los dos subconjuntos.
18. Se leen dos listas de números enteros A y B de 100 y 60 elementos respectivamente. Se desea resolver mediante subprogramas las siguientes tareas:
- 1) Ordenar cada una de las listas A y B .
 - 2) Crear una lista C que contenga la mezcla de las listas ordenadas A y B y visualizarla.
 - 3) Localizar si el número 255 está en la lista C .
19. Se desea almacenar polinomios de coeficientes enteros y una variable real con grado máximo 25 utilizando arrays. Define el tipo adecuado y escribe subprogramas que:
- 1) Dado un punto y un polinomio lo evalúe en dicho punto.
 - 2) Dados dos polinomios obtenga el polinomio suma.
 - 3) Dado un polinomio obtenga el polinomio derivada.
 - 4) Escriba un polinomio por pantalla.
20. La tabla:

x	3.0	3.3	3.6	4.0	4.5
$f(x)$	14.9	18.2	22.3	27.2	33.3

muestra los valores de una función f para ciertos valores de x . Se desea usar estos valores para estimar los valores de f para otros valores de x distintos, dentro del intervalo dado por la tabla. Escribe un programa que almacene la tabla anterior y que para cada valor de x escriba la correspondiente aproximación de $f(x)$.

Nota: Esta aproximación puede ser obtenida mediante el método de *interpolación lineal*. Con este método, se traza una línea entre los dos puntos de la tabla que rodean al valor de x a evaluar. Es decir si los puntos de la tabla posterior y anterior son x_a y x_b , el valor de $f(x)$ viene dado por:

$$f(x) = f(x_b) + \frac{f(x_a) - f(x_b)}{x_a - x_b} \cdot (x - x_b)$$

Supón que quieres evaluar $f(3.5)$, en ese caso los valores de x_a y x_b son $x_a=3.6$ y $x_b=3.3$.

21. Dados N valores de x y $f(x)$ la línea recta que más se acerca a estos valores (según el método de los *mínimos cuadrados*) es:

$y = a + b x$, donde

$$b = \frac{N \cdot \sum_{i=1}^N x_i \cdot f(x_i) - \sum_{i=1}^N x_i \cdot \sum_{i=1}^N f(x_i)}{N \cdot \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2}$$

$$a = \frac{\sum_{i=1}^N f(x_i) - b \cdot \sum_{i=1}^N x_i}{N}$$

Escribe un programa que lea el número total de puntos (N) y los N pares de $(x, f(x_i))$ y calcule los valores de a y b para dichos datos. El programa debe ser válido para 50 o menos datos.

22. Escribir un programa que lea 20 números naturales, encuentre el valor máximo y lo imprima, junto con el número de veces que aparece, y las posiciones en que esto ocurre. Este proceso se debe repetir con el resto de los números hasta que no quede ningún número por tratar.
23. Escribir un programa que lea 42 números enteros en un array de dimensión 7 x 6 y realice las siguientes operaciones:
- 1) Imprimir el array
 - 2) Encontrar el elemento mayor del array
 - 3) Indicar la posición (fila y columna) del elemento mayor
 - 4) Si el elemento mayor está repetido, indicar cuantas veces y la posición de cada elemento repetido.
24. Escribir un programa que calcule la traspuesta de una matriz cuadrada. Utilizar la representación de los ejercicios 6 y 7. La matriz traspuesta se obtiene cambiando filas por columnas.

Nota: Detecta posibles errores en la dimensión de la matriz.

25. Escribe un programa que lea un texto del teclado y determine si es palíndromo o no.

Nota: un texto es palíndromo si se lee igual en ambos sentidos (cuidado con los espacios en blanco, mayúsculas y minúsculas y vocales acentuadas). Ejemplos:

“Salta Lenin el atlas”

“Isaac no ronca así”

“somos o no somos”

26. Escribe un programa que lea un texto del teclado, sustituya todas las secuencias de dos o más espacios en blanco por un único espacio en blanco y escriba el texto resultante por pantalla.
27. Escribe un programa que lea un texto de teclado e imprima en pantalla un mensaje indicando cuantas veces aparecen dos letras contiguas e iguales en el mismo.
28. Se desea efectuar operaciones de resta, multiplicación y división entera y resto de cantidades enteras que se caracterizan por tener una gran cantidad de dígitos, de manera

que queda excluido el uso del tipo REAL. Para ello cada número se representará como una cadena de caracteres. Escriba un subprograma que realice cada una de las operaciones anteriores.

Nota: Se considera que no se va a trabajar con números de más de 100 dígitos.

Como aplicación, escribe un programa que calcule el factorial de 50.

29. Escribe un programa similar al del problema 11, pero que opere con números complejos. Define el tipo complejo como un registro formado por dos números reales que representen la parte real e imaginaria del número. Escribe subprogramas para sumar, restar, multiplicar, dividir, calcular conjugados y comparar números complejos.
30. Escribe una función que tome como parámetro un registro que representa una fecha y devuelva TRUE si esta fecha es válida o FALSE en otro caso.
31. Un vector de registros contiene la descripción de una serie de no más de 100 personas. Cada registro contiene los campos: nombre, edad, sexo y altura. Escribir un programa que lea y almacene datos en un vector de este tipo, ordene el vector por la altura y lo muestre por pantalla.
32. Dada la siguiente definición de tipo:

TYPE

CONJUNTONUMEROS = **SET OF** MINIMO..MAXIMO;

donde MINIMO y MAXIMO son dos constantes naturales, escribe un programa que incluya un procedimiento para imprimir una variable del tipo NUMEROCONJ.

Ejemplo: El conjunto cuyos miembros son 3, 7, 11, 19 deberá imprimirse de la siguiente forma:

{3, 7, 11, 19}

33. Dada la siguiente definición de tipo conjunto:

CIFRAS = 0..9;

CONJUNTO = **SET OF** CIFRAS;

escribe funciones que calculen:

- 1) La cardinalidad de un CONJUNTO.
- 2) Distancia entre dos conjuntos (menor diferencia entre sus elementos):

$$Distancia(A, B) = \min\{|a - b| \mid a \in A, b \in B\}$$

34. Escribe un programa que lea un párrafo de no más de 80 letras y que tenga como salida:
 - 1) El número de vocales que contiene.
 - 2) El número de minúsculas, mayúsculas y dígitos.
 - 3) Frecuencia de cada una de las letras del alfabeto.
35. Los alumnos de una clase desean celebrar una cena de confraternización un día del presente mes en el que puedan asistir todos. Se pide realizar un programa que recoja para cada alumno los días que le vendrían bien para ir a la cena e imprima las fechas concordantes para todos los alumnos. Los datos se introducirán por teclado

Relación de problemas (Tema 7)

1. Escribe un programa que lea del teclado el nombre de tres ficheros de texto y escriba en el tercero el contenido del primero seguido del contenido del segundo.
2. Escribe un programa que codifique y decodifique ficheros de texto usando el método de *cifrado César*. Para codificar un fichero de texto (cuyo nombre y extensión se leerá de teclado) se generará otro fichero de texto con el mismo nombre y con extensión ".COD". La codificación consistirá en reemplazar cada carácter del fichero original por el tercero siguiente según la tabla ASCII (Por ejemplo, el carácter 'a' será sustituido por el carácter 'd'). Si el carácter a codificar es `CHR(10)` o `CHR(13)` no se cambiará al codificarlo. La opción de decodificación deberá leer de teclado el nombre de un fichero codificado y recuperar en otro fichero con mismo nombre y con extensión ".DEC" la información original.

NOTA: El desplazamiento en tres caracteres debe ser circular tanto en un sentido como en otro. Por ejemplo, el último carácter de la tabla ASCII (`CHR(255)`) es codificado como `CHR(2)`, ya que el último carácter de la tabla ASCII tiene código 255 y el primero 0.

3. Codifica un programa que cree un fichero que contenga los datos relativos a los artículos de un almacén. Para cada artículo habrá que guardar la siguiente información:

Código del artículo	(Numérico)
Nombre del artículo	(Cadena de caracteres)
Existencias actuales	(Numérico)
Precio	(Numérico).

Los datos de los distintos artículos se pedirán del teclado y se almacenarán previamente en un array. Supondremos que el número máximo de artículos que se pueden introducir es treinta. Una vez leídos, los datos se ordenarán en el array (el orden será ascendente según el código del artículo) y se grabarán en el fichero.

4. Escribe un programa que a partir de dos ficheros generados por el programa anterior (recordar que los artículos están ordenados por código) genere un tercer fichero que sea el resultado de mezclar de forma ordenada los dos primeros. Escribe otro programa que haga lo mismo pero con n ficheros (n será un número positivo leído de teclado).
5. Escribe un programa que tome como entrada el nombre de un fichero generado por el programa del ejercicio 3 y una condición sobre el campo referente a las existencias actuales. La condición podrá ser:

[<,<=,>,>=, <>, =] <número>

Es decir, que las referencias actuales sean menores a un número dado, o menores o iguales, o mayores, etc.

El programa debe generar como salida un fichero llamado "salida.dat" que contenga todos aquellos artículos para los que se cumple la condición de entrada.

6. Escribe un programa que tenga como entrada el nombre de un fichero que contenga un texto y muestre en pantalla una estadística de la longitud de las distintas palabras que contiene (número de palabras de cada longitud), así como cuántas veces aparecen los caracteres separadores (" ", ".", ":", ";", "¡").
7. Escribe un programa que a partir de un fichero de texto de entrada que contiene números

enteros positivos expresados en cualquier base entre 2 y 10, genere un listado por pantalla y un fichero de salida que contenga las mismas cantidades del fichero de entrada, pero expresadas en base 10.

El fichero de entrada contendrá dos líneas para cada número; la primera tendrá el número propiamente dicho y la segunda la base en que está expresado.

Ejemplo:

Fichero de Entrada:

```
101
2
412
5
111
9
923
10
```

Salida por pantalla: Se deberá generar un listado por pantalla con las siguientes líneas:

Número de entrada	2/101	Resultado	5
Número de entrada	5/412	Resultado	107
Número de entrada	9/111	Resultado	91
Número de entrada	10/923	Resultado	923

El contenido del fichero de salida será:

```
5
107
91
923
```

8. Escribe un programa que busque en un archivo binario que contiene números enteros, los valores máximo y mínimo y los muestre por pantalla.
9. Escribe un programa que calcule el número de líneas que han de leerse de un archivo de texto hasta que se hayan leído del fichero todas las letras minúsculas del alfabeto al menos una vez. El programa debe mostrar también el número total de caracteres leídos hasta ese momento. Si se alcanza el final del fichero y no se consigue el objetivo, el programa mostrará un mensaje por pantalla que lo indique.
10. Escribe un programa que a partir de un fichero de texto, cree otro que sólo contenga aquellas líneas del primero en las que figuren una palabra introducida previamente por el usuario desde el teclado.

NOTA: Suponer que la longitud máxima de una línea del fichero es 128 caracteres y que las palabras están separadas por espacios en blanco o saltos de línea y que los únicos caracteres que aparecen en el texto son letras y espacios en blanco.

11. Escribe un programa que mezcle dos archivos binarios que contengan números enteros separados por espacios, y que estén ordenados ascendentemente. El fichero resultante deberá estar también ordenado en orden ascendente.
12. Escribe un programa que a partir de los nombres de dos ficheros binarios que contienen valores cardinales en el rango [1..1000], genere dos nuevos ficheros con la unión e intersección de los dos ficheros anteriores.
13. Se desea escribir un programa que resuelva el juego *Sopa de Letras*. Para ello, se considerarán las siguientes especificaciones:
 - La tabla de caracteres que forma la sopa de letras tendrá como dimensiones 15 filas por 15 columnas.
 - En fichero de entrada "SOPA.DAT" contendrá las distintas líneas que forman la tabla.
 - En fichero "PATRONES.DAT" contendrá las palabras a buscar dentro de la SOPA.
 - Sólo será necesario buscar los patrones dentro de la SOPA por filas y por columnas en ambos sentidos (No será necesario buscar en las diagonales).

La salida del programa será un listado por pantalla mostrando cada palabra encontrada junto con las coordenadas donde fue hallada.

14. Mejorar el programa anterior para que busque también en diagonales
15. Se desea ordenar un fichero de números enteros que no cabe en la memoria central del ordenador. Se opta por ordenarlo considerándolo como un array en disco. Para ello será necesario escribir los siguientes subprogramas:
 - 1) **PROCEDURE** Longitud (NombreFich : NOMBREFICH) : **CARDINAL**; Devuelve la cantidad de números enteros almacenada en el fichero que toma como argumento.
 - 2) **PROCEDURE** LeerFich (NombreFich : NOMBREFICH; Posicion : **CARDINAL**) : **INTEGER**; Devuelve el valor del número que está situado en la posición indicada dentro del fichero.
 - 3) **PROCEDURE** EscribirFich (NombreFich : NOMBREFICH; Posicion : **CARDINAL**; Valor : **INTEGER**); Modifica el contenido del fichero, de modo que el valor del número que ocupa la posición indicada pasa a ser Valor. El resto de los números en el fichero permanecen iguales. Para ello debe crearse un fichero auxiliar a partir del original y el nuevo valor y copiar éste sobre el original.

Utilizando estos subprogramas, diseña el programa que ordene el fichero y muestre el resultado por pantalla.

16. Otra alternativa al procedimiento anterior es dividir el fichero inicial en subficheros de como máximo 50 números. Cada uno de estos ficheros se carga en memoria en un vector de tamaño 50, se ordena en el vector y se vuelca en el correspondiente subfichero. Posteriormente se mezclan los subficheros obteniéndose el fichero original ordenado. Escribe un programa para esto.

Nota: Los problemas 14, 15 y 16 presentan un alto nivel de dificultad.

