

Diseñar un algoritmo que lea 3 números e imprima el mayor y el menor valor de los tres. La

```
ALGORITMO MAXMIN_3;
    entero A, B, C, MAX, MIN;
INICIO
    LEER A, B, C;
    SI A < B ENTONCES
        SI A < C ENTONCES
            MIN = A;
            SI B > C ENTONCES
                MAX = B;
            SINO
                MAX = C;
            FIN SI;
        SINO
            MIN = C;
            MAX = B;
        FIN SI;
    SINO
        SI A > C ENTONCES
            MAX = A;
            SI B > C ENTONCES
                MIN = C;
            SINO
                MIN = B;
            FIN SI;
        SINO
            MIN = B;
            MAX = C;
        FIN SI;
    FIN SI;
    ESCRIBIR "El mayor es: ", MAX
    ESCRIBIR "El menor es: ", MIN
FIN.
```

Diseñar un algoritmo para leer una lista de números positivos y escribir, despues de leerla, el valor máximo y el mínimo de esa lista. El algoritmo leerá un número tras otro hasta que se introduzca un número negativo. Es decir, el algoritmo dejará de leer números cuando lea un número negativo y, entonces, mostrará los valores resultantes.

```
ALGORITMO MAXMIN_Lista;
    entero N, MAX, MIN;
INICIO
    LEER N;
    SI (N < 0) ENTONCES
        ESCRIBIR "Lista vacía:
                    No hay máximo ni mínimo."
    SINO
        MAX = N;
        MIN = N;
        LEER N;
        MIENTRAS (N >= 0) HACER
            SI N>MAX ENTONCES
                MAX = N;
            SINO
                SI (N < MIN) ENTONCES
                    MIN = N;
                FIN SI.
            FIN SI;
            LEER N;
        FIN MIENTRAS;
        ESCRIBIR "Mayor: ", MAX
        ESCRIBIR "Menor: ", MIN
    FIN SI;
FIN.
```

Diseñar un algoritmo que lea un texto carácter a carácter y reemplace cada cadena de uno o más espacios en blanco por un sólo espacio. Es decir, en la salida no se deben escribir más de un espacio seguido. Termina cuando encuentra el carácter '.' (punto).

```
ALGORITMO Quita_espacios;
    caracter CH_ANT, CH_ACT;
INICIO
    LEER CH_ACT; /
    CH_ANT = 'x' /*Asignar al anterior cualquiera != ' '*/
    /* Mientras no lleguemos al final... */
    MIENTRAS (CH_ACT != '.') HACER
        /* Escribir CH_ACT si no se cumple que los dos sean espacios */
        SI NOT (CH_ACT == ' ') AND
            (CH_ANT == ' ')
        ENTONCES
            ESCRIBIR CH_ACT;
        FIN SI;
        /* Asig. al carácter anterior el actual y leemos uno nuevo */
        CH_ANT = CH_ACT;
        LEER CH_ACT;
    FIN MIENTRAS;
FIN.
```

Suponga que la CEE decide poner un límite en los Kg. de pesca que puede recoger en un mes un barco pesquero. Desarrolle un algoritmo al que primero se le dé el límite máximo permitido por la ley y luego se le vayan dando los pesos de lo que se va pescando. Cada vez que se introduzca un nuevo peso se mostrará el total de Kg. pescados hasta ese momento. Cuando se supere el máximo permitido se debe dar la alarma y terminar el algoritmo, indicando el total de Kg. pescados y el número de Kg. en los que se ha sobrepasado el límite legal. El algoritmo también terminará si introducimos un 0 como Kg. pescados y deberá indicar un error si se introduce un número negativo.

**ALGORITMO Pesca;**

```
    real Total, Límite, Pescado;  
INICIO  
    Total = 0;  
    LEER Límite;  
    LEER Pescado;  
    MIENTRAS (Pescado != 0.0) HACER  
        Total = Total + Pescado;  
        SI (Total >= Límite) ENTONCES  
            ESCRIBIR "Límite superado en ",  
                    Total - Límite,  
                    "Kgs."  
        Pescado = 0.0;  
    SINO  
        ESCRIBIR "Total pescado: ",  
                Total  
        LEER Pescado  
    FIN SI.  
    FIN MIENTRAS;  
FIN.
```

Encontrar los n primeros números primos. El número n será leído al principio.

```

ALGORITMO N_Primos;
  entero N, cont, P, I;
  booleano ES_PRIMO;
INICIO
  LEER N;
  SI (N > 0) ENTONCES
    ESCRIBIR 1; /* Escribir el primer primo */
    cont = 1; /* Número de primos que tenemos */
    P = 2; /* Número que vamos a ver si es primo */

    /* Mientras el número de primos calculado sea distinto de N */
    MIENTRAS (cont != N) HACER
      /* Empezamos mirando a partir de la mitad, hacia abajo */
      I = P / 2;
      ES_PRIMO = TRUE; /* Sup. que P es primo */

      MIENTRAS (I>= 2) AND ES_PRIMO HACER
        /* Si I divide a P entonces NO es primo */
        SI (P % I == 0) ENTONCES
          ES_PRIMO = FALSE;
        FIN SI;
        I = I-1; /* Restamos 1 para ver si es divisor de P */
      FIN MIENTRAS;

      /* Si P es primo, lo escribimos y contamos 1 más */
      SI ES_PRIMO ENTONCES
        ESCRIBIR P
        cont = cont + 1;
      FIN SI;
      P = P + 1;
    FIN MIENTRAS;
  FIN SI;
FIN.
```

Diseñar un algoritmo para calcular el N-ésimo término de la sucesión de Fibonacci, donde N es un entero positivo. La sucesión de Fibonacci está definida por:

```
Fibonacci (0) = 0
Fibonacci (1) = 1
Fibonacci (n) = Fibonacci (n-1) + Fibonacci (n-2)
```

**ALGORITMO Fibonacci;**  
**entero N, F1, F2, i;**

**INICIO**

**LEER N;** /\* Leer el número del término \*/

**SI (N == 0) ENTONCES**

**ESCRIBIR 0**

**SINO**

**F1 = 0;** /\* Los dos primeros términos \*/

**F2 = 1;**

**PARA (i:=2; i<=N; i++) HACER**

**F2 = F1 + F2;** /\* Calcular nuevo término \*/

**F1 = F2 - F1;** /\* Término anterior \*/

**FIN PARA;**

**ESCRIBIR F2;**

**FIN SI;**

**FIN.**

Dentro del bucle **PARA**, también valdría lo siguiente, donde **Aux** es una variable Natural, auxiliar para efectuar el intercambio :

**Aux = F2;**

**F2 = F1 + F2;** /\* Calcular nuevo término \*/

**F1 = Aux;** /\* Término anterior \*/