

IES Celia Viñas - 1º ASI – Fundamentos de programación

**Una solución modular para el ejercicio obligatorio de programación estructurada:
convertir números a letras**

algoritmo escribir_números

variables

numero es entero
unid, dec, cent son enteros /* Unidades, decenas, centenas */
umill, dmill son enteros /* Unidades de millar y decenas de millar*/

inicio

escribir("Introduzca un número entero (máximo 5 cifras)")
leer(numero)
extraer_digitos(numero, &unid, &dec, ¢, &umill, &dmill) /* Separar el nº en dígitos */
escribir_dec_millar(umill, dmill)
escribir_uni_millar(umill, dmill)
escribir_centenas(unid, dec, cent)
escribir_decenas(unid, dec)
escribir_unidades(unid, dec)

fin

/* Extraer los dígitos del número mediante divisiones sucesivas */

procedimiento extraer_digitos(numero es entero, *U es entero, *D es entero,
 *C es entero, *UM es entero, *DM es entero)

inicio

DM = numero **div** 10000 /* Extraer decenas de millar */
numero = numero - (DM * 10000) /* Extraer unidades de millar */
UM = numero **div** 1000
numero = numero - (UM * 1000) /* Extraer centenas */
C = numero **div** 100
numero = numero - (C * 100) /* Extraer decenas */
D = numero **div** 10
numero = numero - (D * 10) /* Extraer unidades */
UM = numero

fin

/* Escribir las decenas de millar. Se escriben igual que las decenas convencionales */

procedimiento escribir_dec_millar(umill es entero, dmill es entero)

inicio

escribir_decenas(umill, dmill) /* Reutilizamos la función escribir_decenas */

fin

/* Escribir unid. de millar. Se escriben como las unidades convencionales, seguidas de la palabra "mil". Hay varias excepciones. Cuando la unidad de millar es 1 se escribe diferente si hay decenas de millar ("un mil" → 31050) que si no las hay ("mil" → 1050). En otro caso (30050, 32050), se escribe la unidad seguida de la palabra "mil" */

procedimiento escribir_uni_millar(umill es entero, dmill es entero)

inicio

si (umill == 1) **entonces** /* Casos especiales */
 inicio
 si (dmill > 1) **entonces** /* 21xxx, 31xxx, 41xxx, etc... */
 escribir(" un mil ")
 si_no
 escribir(" mil ") /* 11xxx, 1xxx */
 fin
 si_no /* Casos generales */
 inicio
 escribir_unidades(umill, dmill)
 si (umill !=0) **y** (dmill !=0) **entonces**
 escribir(" mil ")
 fin

fin

/* Escribe las centenas. La única excepción es 100 */

procedimiento escribir_centenas(unid es entero, dec es entero, cent es entero)

inicio

```

si (unid == 0) y (dec == 0) y (cent == 1) entonces      /* Caso excepcional */
    escribir ("cien")
si_no                                              /* Caso general */
    según (cent) hacer
    inicio
        1: escribir("ciento")
        2: escribir("doscientos")
        3: escribir("trescientos")
        ...
        9: escribir("novecientos")
    fin
fin

/* Escribir las decenas. Los números del 10 al 19 se tratan como excepciones. También el 20
frente al caso general (2x = "veinti..."). En el resto de decenas, se escribe una " y " si la
unidad no es cero (ej: 3x = "treinta y ...") */
procedimiento escribir_decenas(unid es entero, dec es entero)
inicio
    si (dec == 1) entonces                          /* Casos especiales (de 10 a 19) */
        según (unid) hacer
        inicio
            0: escribir("diez")
            1: escribir("once")
            2: escribir("doce")
            3: escribir("trece")
            ...
            9: escribir("diecinueve")
        fin
    si (dec == 2) entonces                          /* De 20 a 29 */
    incio
        si (unid == 0) entonces
            escribir ("veinte")
        si_no
            escribir ("veinti")
    fin
    si (dec >= 3) entonces                          /* Resto de casos (de 30 a 99) */
    inicio
        según (dec) hacer
        inicio
            3: escribir("treinta")
            4: escribir("cuarenta")
            5: escribir("cincuenta")
            ...
            9: escribir("noventa")
        fin
        si (unid != 0) entonces
            escribir (" y ")
    fin
fin /* procedimiento escribir_decenas */

/* Escribe las unidades. Excepción: cuando la decena es 1 (10, 11, 12, etc). Estos casos
se escriben en el procedimiento escribir_decena */
procedimiento escribir_unidades(unid es entero, dec es entero)
inicio
    si (dec != 1) entonces                          /* Si el n° está entre 10 y 19, no escribir nada */
        según (unid) hacer                          /* En cualquier otro caso, escribir la unidad */
        inicio
            1: escribir("uno")
            2: escribir("dos")
            3: escribir("tres")
            ...
            9: escribir("nueve")
        fin
    fin
fin

```