

RUTINA MONTO ESCRITO

Por: Roberto Flórez R.

Departamento de Ingeniería de Sistemas

Facultad de Ingeniería. Universidad de Antioquia

La siguiente es una rutina llamada MONTO-ESCRITO, implementada en COBOL, la cual es de gran utilidad, principalmente en aplicaciones de nómina, donde se requiere que el computador imprima los cheques de pago de los trabajadores.

Del programa de liquidación se obtiene el valor neto a pagar a cada trabajador. Los cheques requieren que el valor aparezca tanto en números como en letras. Dicha rutina recibe como dato el valor a pagar y lo convierte a su equivalente en letras. Esta rutina convierte valores desde 1 centavo hasta novecientos noventa y nueve mil novecientos noventa y nueve pesos con 99 centavos.

El algoritmo consiste en dividir el valor leído en grupos de tres cifras, sin considerar los centavos, y hallar el va-

lor en letras para ese grupo de tres cifras. Así, por ejemplo, si se tiene el valor 398.398.55 habrá dos grupos de tres cifras: 398 y 398, la única diferencia será que cuando se procese el primer grupo se debe agregar la palabra MIL.

Se definen arreglos (vectores) en los cuales se incluyen todas las palabras posibles para expresar cantidades con palabras. Habrá por lo tanto un arreglo para las palabras de las centenas, otro para las de las decenas, otro para las de las unidades y otro para palabras especiales. Asociado con cada uno de estos arreglos habrá otro que indica la posición del arreglo en la cual empieza una determinada palabra. Así por ejemplo el arreglo llamado CENTEN, que contiene las palabras CIEN-TO, DOSCIENTOS, TRESCIENTOS, etc., tiene un arreglo asociado llamado

LONGC que contiene las posiciones desde donde empieza cada palabra; entonces 001 indica que la palabra CIENTO empieza en la posición uno del arreglo CENTEN, 008 indica que la palabra DOSCIENTOS empieza en la posición 8 del arreglo CENTEN, 019 indica que la palabra TRESCIENTOS empieza en la posición 19 del arreglo CENTEN, y así sucesivamente.

El valor en letras se escribe en un arreglo llamado SALIDA, el cual tiene 84 posiciones. Dicho arreglo se llena con asteriscos al comienzo de cada valor.

Las variables I,J,K,L,M, VECES e Y son variables controladoras de ciclos. La variable NETO contiene el valor a ser trasladado a letras.

DATA DIVISION

WORKING-STORAGE SECTION.

77 I PIC 999 VALUE ZEROS.

77 J PIC 999 VALUE ZEROS.

77 K PIC 999 VALUE ZEROS.

77 L PIC 999 VALUE ZEROS.

77 M PIC 999 VALUE ZEROS.

77 VECES PIC 999 VALUE ZEROS.

77 Y PIC 999 VALUE ZEROS.

77 PP PIC 9 VALUE ZERO.

01 ANAL.

02 UDC PIC 999 VALUE ZEROS.

01 ANAL 1 REDEFINES ANAL.

02 DIGITO OCCURS 3 TIMES PIC 9.

01 OUT.

02 SALIDA OCCURS 84 TIMES PIC X.

01 NN1.

02 NETO PIC 9(8).

01 NN2 REDEFINES NN1.

02 PER1 PIC 999.

02 PER2 PIC 999.

02 CVS1 PIC 9.

02 CVS2 PIC 9.

01 CENTEN

02 FILLER PIC X(104) VALUE "CIENTO DOSCIENTOS TRESCIENTOS CUATROCIENTOS QUINIENTOS SEISCIENTOS SETECIENTOS OCHOCIENTOS NOVECIENTOS".

01 LEO2 REDEFINES CENTEN.

02 LETRA1 OCCURS 104 TIMES PIC X.

01 LONGC.

02 LON1 PIC X(30)

VALUE "001008019031045056068080092104".

01 LONG1 REDEFINES LONGC.

02 DCENT OCCURS 10 TIMES PIC 999.

01 DEDEN.

02 FILLER PIC X(71) VALUE "DIECIVEINTITREINTA CUARENTA CINCUENTA SESENTA SETENTA OCHENTA NOVENTA".

01 LEO4 REDEFINES DEDEN.

02 DDECE OCCURS 71 TIMES PIC X.

01 LONGD.

02 FILLER PIC X(30)

VALUE "001006012020029039047055063071".

01 LON1D REDEFINES LONGD.

02 LARGOE OCCURS 10 TIMES PIC 999.

01 DECE1.

02 FILLER PIC X(73) VALUE

"DIEZ ONCE DOCE TRECE
 CATORCE QUINCE VEINTE
 UNO CERO CIEN MIL CON
 00/100Y ".
 01 LEO6 REDEFINES DECE1.
 02 LIODE OCCURS 73 TIMES
 PIC X.
 01 DELONG.
 02 FILLER PIC X (42)
 VALUE "00100601101602
 2030037044048053058062
 072073".
 01 RED REDEFINES DELONG.
 02 MAEHP OCCURS 14 TIMES
 PIC 999.
 01 UNIDAD.
 02 FILLER PIC X (48) VALUE
 "UN DOS TRES CUATRO CIN-
 CO SEIS SIETE OCHO NUE-
 VE ".
 01 LE12 REDEFINES UNIDAD.
 02 LET3 OCCURS 48 TIMES
 PIC X.
 01 LONGUI.
 02 FILLER PIC X (30)
 VALUE "0010040080
 130200260310370420
 48".
 01 RLONG REDEFINES LONGUI.
 02 LOLO OCCURS 10 TIMES
 PIC 999.
 PROCEDURE DIVISION.
 MONTO.
 MOVE ZEROS TO K, PP
 ACCEPT NETO LINE 1 POSI-
 TION 10 ERASE PROMPT
 ECHO.
 PERFORM LLENAS UNTIL K
 = 84.
 OTRA.
 ADD 1 TO VECES.
 MOVE PER2 TO UDC.

MOVE ZEROS TO I, J, K.
 IF VECES = 1 MOVE PER1 TO
 UDC.
 IF DIGITO(1) = 0 GO TO RU-
 DECE.
 IF UDC = 100 MOVE " " TO
 LETRA1(5).
 MOVE " " TO
 LETRA1(6).
 MOVE DIGITO(1) TO I, J.
 MOVE DCENT(I) TO I, K.
 ADD 1 TO J.
 MOVE DCENT(J) TO J.
 PERFORM CENTE UNTIL
 K = J.
 MOVE "T" TO LETRA1(5)
 MOVE "O" TO LETRA1(6)
 RUDECE.
 IF DIGITO(2) = 0 GO TO
 RUNID.
 IF DIGITO(2) = 1 AND DI-
 GITO(3) > 5
 PERFORM COMO1
 GO TO RUNID.
 IF DIGITO(2) = 1
 ADD 1 TO DIGITO(3)
 MOVE DIGITO(3) TO L
 MOVE DIGITO(3) TO M
 ADD 1 TO M
 PERFORM COMO3
 ELSE
 IF DIGITO(2) = 2
 AND
 DIGITO(3) = 0
 MOVE 7 TO L
 MOVE 8 TO M
 PERFORM COMO3
 ELSE GO TO CUALD.
 IF VECES = 1 GO TO
 PONMIL.
 GO TO CON.
 CUALD.

PERFORM COMO1.
RUNID.
IF VECES = 2 GO TO RUNISI.
IF UDC = 0 GO TO OTRA.
IF DIGITO(3) = 0 GO TO
PONMIL.
PERFORM COMO2.
GO TO PONMIL.

RUNISI.
IF DIGITO(3) = MOVE 8 TO L
MOVE 9 TO M
PERFORM
COMO3
GO TO CON.
IF DIGITO(3) = 0 GO TO UNI-
SO.
PERFORM COMO2.

CON.
MOVE 12 TO L.
MOVE 13 TO M.
PERFORM COMO3.
SUBTRACT 5 FROM Y.
PERFORM CTVOS.
MOVE ZEROS TO VECES, NE-
TO, K, Y.
DISPLAY OUT LINE 2 POSI-
TION 1.
GO TO FIN.

UNISO.
IF NETO NOT = 0 GO TO CON.
MOVE 9 TO L.
MOVE 10 TO M.
PERFORM COMO3.
GO TO CON.

PONMIL.
MOVE 11 TO L.
MOVE 12 TO M.
PERFORM COMO3.
GO TO OTRA.

LLENAS.
ADD 1 TO K.
MOVE "*" TO SALIDA(K).

CENTE
ADD 1 TO Y.
MOVE LETRA1 (K) TO SALI-
DA (Y).
ADD 1 TO K.

DECENA
ADD 1 TO Y.
MOVE DDECE(K) TO SALIDA
(Y).
ADD 1 TO K.

DECEN1.
ADD 1 TO Y.
MOVE LIODE(K) TO SALIDA
(Y).
ADD 1 TO K.

DESCU.
ADD 1 TO Y.
MOVE LET3(K) TO SALIDA
(Y).
ADD 1 TO K.

COMO3.
MOVE MAEHP (L) TO I, K.
MOVE MAEHP (M) TO J.
PERFORM DECEN1 UNTIL K
= J.

CTVOS.
MOVE CVS1 TO SALIDA (Y)
ADD 1 TO Y
MOVE CVS2 TO SALIDA (Y).

COMO2.
MOVE DIGITO(3) TO I, J.
MOVE LOLO(I) TO I.
ADD 1 TO J.
MOVE LOLO(J) TO J.
MOVE I TO K.
PERFORM DESCU UNTIL K
= J

COMO 1
MOVE DIGITO(2) TO I, J.
MOVE LARGOE(I) TO I, K.
ADD 1 TO J.
MOVE LARGOE(J) TO J.

PERFORM DECENA UNTIL
K = J.
IF DIGITO(3) NOT = 0 AND
DIGITO(2) > 2
ADD 1 TO Y

MOVE "Y" TO SALIDA (Y)
ADD 1 TO Y
MOVE "" TO SALIDA (Y).
FIN.
STOP RUN.
END PROGRAM.