org $20000                     TEST INPUT DATA
inval  dc.l  $1234abcd        Input Value (will be "printed" in hex)
outstr ds.b  8               String Where the Hex Representation will be "printed" in
                                Once done this will contain '1','2','3','4','a','b','c','d'
                                '1' is the character 1 (49 in decimal)

org $30000

main
movea.l #inval,-(a7)        Push address of inval onto the stack
movea.l #outstr,-(a7)       Push starting address of string
jsr prnhex                  Call prnhex
                                jsr is a bsr that can jump further away than +/-32K
trap  #15

org $31000

prnhex
movea.l d4, -(a7)           Save registers we will be touching
move.a.l d3, -(a7)
movea.l d2, -(a7)
movea.l d1, -(a7)
movea.l a0, -(a7)
movea.l 24(a7), a1          Get pointer to output string in a1
movea.l 28(a7), a0          Get pointer to input value in a0
movea.l (a0), d2            Read input value in d2
movea.l #$f0000000, d3      This number is 11110000...0
                                If we AND this with the input number we get the most
                                significant hex digit
movea.l #28, d0             If we shift the most significant digit RIGHT by 28 bits
                                right we move the digit down to rightmost position
                                Hence we convert the digit to a number between 0 and 15
ploop  movea.l d2, d4        D4 = input value
and.a.1 d3, d4              AND with digit mask
lsr.l d0, d4       SHIFT right so that the digit starts from the zero position
SEE BELOW FOR NOTES ON THIS CODE
this way it is a number between 0 and 15
add.l #'0', d4    Adding the number ‘0’ (character 0) converts numbers 0 to 9
to characters ‘0’ through ‘9’ respectively
cmp.l #'9', d4    So long as the resulting character is less or equal ‘9’ we
are done
ble isdigit
sub.l #'0', d4    number is between a and f
sub.l #10, d4    convert it to 0 through 5
add.l #'a', d4    adding ‘a’ converts it to ‘a’ through ‘f’
isdigit move.b d4,(a1)+ d4 is the next hex digit, copy to the output string
subq.l #4, d0    Decrement the shift-right count by 4 so that we
appropriately shift the next digit
lsr.l #4, d3    Shift digit mask by 4 to point to the next HEX digit
bne ploop      Continue so long as there are more digits
Eventually the Digit Mask will become 0.
move.l (a7)+,a0 Restore registers
move.l (a7)+,d1
move.l (a7)+,d2
move.l (a7)+,d3
move.l (a7)+,d4
rts Return

NOTES:

D3 is the DIGIT MASK
D0 is the SHIFT RIGHT AMOUNT

We start with the following to extract the leftmost hex digit (upper 4 bits)
D3 = $f000 0000
D0 = 28
If d4 is holding the input number 1234abcd then
   AND D3, D4 → D4 = $1000 0000
   LSR.L D0, D4 → D4 = $0000 0001
Second iteration: Next hex digit
D3 = $0f00 0000
D0 = 24
   AND D3, D4 → D4 = $0200 0000
   LSR.L D0, D4 → D4 = $0000 0002
Third iteration
D3 = $00f0 0000
D0 = 20
   AND D3, D4 → D4 = $0030 0000
   LSR.L D0, D4 → D4 = $0000 0003
Fourth iteration
D3 = $000f 0000
D0 = 16
   AND D3, D4 → D4 = $0004 0000
   LSR.L D0, D4 → D4 = $0000 0004
Fifth iteration
D3 = $0000 f000
D0 = 12
   AND D3, D4 → D4 = $0000 a000
   LSR.L D0, D4 → D4 = $0000 000a
Sixth iteration
D3 = $0000 0f00
D0 = 8
   AND D3, D4 → D4 = $0000 0b00
   LSR.L D0, D4 → D4 = $0000 000b
Seventh iteration
D3 = $0000 00f0
D0 = 4
   AND D3, D4 → D4 = $0000 00c0
LSR.L D0, D4 → D4 = $0000 000c

Final iteration
D3 = $0000 000f
D0 = 0
   AND D3, D4 → D4 = $0000 000d
LSR.L D0, D4 → D4 = $0000 000d

MOVEM Instruction:
You can replace the following instructions:
move.l  d4, -(a7)       Save registers we will be touching
move.l  d3, -(a7)
move.l  d2, -(a7)
move.l  d1, -(a7)
move.l  a0, -(a7)
with:
  movem.l  d1-d4/a0,-(a7)
Similarly:
  movem.l  (a7)+,d1-d4/a0
replaces:
move.l  (a7)+,a0       Restore registers
move.l  (a7)+,d1
move.l  (a7)+,d2
move.l  (a7)+,d3
move.l  (a7)+,d4

NOTES on how to covert a digit to a character
Characters ‘0’ through ‘9’ are characters 48 through 57
Characters ‘a’ through ‘f’ are characters 65 through 69
To print a digit 0 through 15 to a hex digit we have to map
Numbers 0 through 9 to characters ‘0’ through ‘9’ and numbers
10 through 15 to characters ‘a’ through ‘f’
we do the following calculation
\begin{verbatim}
N = N + '0'
If (N > '9') THEN
    N = N + 'a' - '0' - 10;
\end{verbatim}