

```

;Computer Architectures A.Y. 2012/2013
;Riccardo Ierardi s194426
;Self Test
;riccardo.ierardi@gmail.com
.MODEL small
.STACK
.DATA
INPUT_BIN DW ?
OUTPUT_BCD DB 5 DUP(?)
.CODE
.STARTUP

    MOV INPUT_BIN,8081H    ;"Random" initialization of input variable,just for test
    MOV AX, INPUT_BIN
    XOR DX,DX             ;Storing converted values directly in DX and AL
    SHL AX,1              ;First 2 shifts cannot create a nibble higher than 5
    RCL DX,1              ;so no need to check
    SHL AX,1
    RCL DX,1
    SHL AX,1
    RCL DX,1
    CMP DX,5              ;The third could create it
    JB NO_ADD_0
    ADD DX,3
NO_ADD_0:    MOV CX,12     ;Setting up the loop, now after each shift a check is
needed
SHIFT_LOOP: SHL AX,1      ;In total we need 16 shift, but 3 have been already done
and the last
    RCL DX,1              ;is at the end because it must not be checked
    ADC AL,0              ;Using the lowest part of AL to store the highest part
of the number
    XOR BL,BL             ;BL temporary register on which perform tests on bits
    MOV BL,DL
    AND BL,00001111B      ;Isolating lowest nibble
    CMP BL,5
    JB TEST_1
    ADD DL,3
TEST_1:     MOV BL,DL
    SHR BL,1              ;Performed 4 SHR because MASM complains for SHL BL,4
    SHR BL,1              ;In this way I isolate the highest nibble of DL and I'm
able to perform
    SHR BL,1              ;the compare with 5
    SHR BL,1
    CMP BL,5
    JB TEST_2
    ADD DL,00110000B      ;With this I can add 3 only to the correct nibble
TEST_2:     MOV BL,DH
    AND BL,00001111B      ;Same as before but with DH
    CMP BL,5
    JB TEST_3
    ADD DH,3
TEST_3:     MOV BL,DH
    SHR BL,1
    SHR BL,1
    SHR BL,1
    SHR BL,1
    CMP BL,5
    JB TEST_4

```

```
TEST_4:  ADD DH,00110000B      ;Same as before but with DH
        MOV BL,AL          ;Perform check on the highest part of the number
        AND BL,00001111B
        CMP BL,5
        JB LOOPING
        ADD AL,3
LOOPING: DEC CX
        JNZ SHIFT_LOOP
        SHL AX,1           ;Here is last shift
        RCL DX,1
        ADC AL,0
        MOV OUTPUT_BCD,DL  ;Storing result in little endian
        AND OUTPUT_BCD,00001111B;I prefer storing values without loop in order to be
able to
        SHR DL,1           ;manage nibbles in a correct way
        SHR DL,1
        SHR DL,1
        SHR DL,1
        MOV OUTPUT_BCD+1,DL
        MOV OUTPUT_BCD+2,DH
        AND OUTPUT_BCD+2,00001111B
        SHR DH,1
        SHR DH,1
        SHR DH,1
        SHR DH,1
        MOV OUTPUT_BCD+3,DH
        MOV OUTPUT_BCD+4,AL

.EXIT
END
```