

```

;
; Este programa escreve o conteúdo da variável inteiro em hexadecimal no ecran
;
;
; assumir que os segmentos têm estes valores
assume cs:cseg, ds:dseg, ss:sseg

; este é o segmento de código
cseg segment

inicio: ; o programa começa aqui
mov ax, dseg
mov ds, ax ; iniciar o DS (Data Segment) com o valor base do segmento de Dados

mov bx, inteiro ; o registo bx fica iniciado com o valor da variável inteiro

mov ax, bx ; coloca em ax o valor de bx
mov cl, 4 ; coloca 4 no registo cl
ror ax, cl ; desloca o registo ax de 4 bits para a direita (se ax=1234h => ax=0123h)
and ah, 0Fh ; assegurar que o registo ah fica apenas com 4 bits ( ah=01h al=23h)
cmp ah, 09 ; compara ah com 9
jg h1 ; se for maior do que 9 então salta para h1
add ah, '0' ; como é menor do que 9, converte-se o número em caracter ( 1 -> '1')
jmp c1 ; agora salta para o local onde se escreve o caracter
h1:
add ah, 'A'- 10 ; como era maior do que 9, é preciso converter (se ah=12 -> ah='C')
c1:
mov dl, ah ; dl recebe o caracter a ser escrito
mov ah, 02h ; código para escrever um caracter no monitor
int 21h ; chamar o DOS para executar a função pretendida

```

; agora continua para o dígito seguinte

```
    mov ax, bx          ;
    and ax, 0F00h      ;
    cmp ah, 09         ;
    jg h2              ;
    add ah, '0'        ;
    jmp c2             ;
h2:
    add ah, 'A'- 10    ;
c2:
    mov dl, ah         ;
    mov ah, 02h       ;
    int 21h           ;

    mov ax, bx        ;
    mov cl, 4         ;
    ror ax, cl        ;
    and al, 0Fh       ;
    cmp al, 09        ;
    jg h3             ;
    add al, '0'       ;
    jmp c3            ;
h3:
    add al, 'A'-10    ;
c3:
    mov dl, al        ;
    mov ah, 02h       ;
    int 21h           ;
```

```

    mov ax, bx          ;
    and ax, 0Fh        ;
    cmp al, 09         ;
    jg h4              ;
    add al, '0'        ;
    jmp c4             ;
h4:
    add al, 'A'-10     ;
c4:
    mov dl, al         ;
    mov ah, 02h        ;
    int 21h           ;

fim:                    ; enquanto não aprendermos de outro modo, terminamos assim :)
    jmp fim

                        ; exit to DOS

    mov ax, 4C00h
    int 21h

cseg ends           ; aqui acaba o segmento de código

dseg segment byte   ; este é o segmento de dados
    inteiro dw 0980Ah ; este é o valor a ser escrito no monitor. Coloca aqui um valor decimal!
dseg ends         ; aqui acaba o segmento de dados

sseg segment stack    ; este é o segmento de stack, por enquanto não nos vamos preocupar com o Stack
    db 100h dup(?)    ; isto define um array com 100h bytes (256) que não estão iniciados (valores desconhecidos)
sseg ends             ; aqui acaba o segmento de dados

end inicio         ; aqui no fim de tudo é que dizemos onde começa a primeira instrução deste programa :)

```