

```

1 ;Inizializza due dati N1 e N2 a 8 bit in memoria
2 ;Se N1<N2 allora ---> Ris=N1*N2
3 ;Se N1>N2 allora ---> Ris(parte bassa)=quoz_intero (N1/N2)
4 ;                               Ris(parte alta) =Resto (N1/N2)
5 ;Se N1=N2 allora ---> Ris=N1+N2
6 ;In tutti i casi il risultato deve essere scritto in memoria
7 ;usando un una stessa variabile a 16 bit
8 ;=====
9 ; p258_Es8.asm
10 ;=====
11 .MODEL SMALL
12
13 .DATA                ;direttiva che dichiara il segmento Dati
14     N1 DB 150
15     N2 DB 017h
16     Ris DW ?        ;alla fine del programma conterrà il risultato
17
18 .STACK              ;direttiva che dichiara il segmento STACK
19
20 .CODE               ;direttiva che dichiara l'inizio del segmento Codice
21 Inizio:
22                 ;Inizializzazione dei registri di segmento
23     MOV AX,@DATA   ;in AX viene posto l'indirizzo del segmento
24     MOV DS,AX      ;Dati per poterlo assegnare al registro DS
25
26 ;programma
27     MOV AL,N1      ;Si assegna al registro AL il valore contenuto nella variabile N1
28
29 ;-- Inizio SE
30     CMP AL,N2      ;Si confronta AL con il contenuto di N2
31     JB  Moltiplica ; --- Se AL<N2 ==> si va all'etichetta Moltiplica
32     JA  Dividi     ; --- Se AL>N2 ==> si va all'etichetta Dividi
33 ; --- Se non è nessuno dei due precedenti (AL=N2) ==> Esegue l'addizione
34     MOV AH,0       ;Si azzerava AH per poi recuperare l'eventuale bit di overflow
35                 ;della somma a 8 bit
36     ADD AL,N2      ;Somma ad AL il valore N2 e se durante la somma
37                 ;si ha un overflow viene settato a 1 il flag CF
38     ADC AH,0       ;AH = AH + 0 + valore del flag di carry CF
39                 ;Dopo questa operazione in AX c'è il risultato della somma
40     MOV Ris,AX     ;che viene portato in Ris
41     JMP Fine_Se    ;infine si salta all'etichetta Fine_se
42
43 Moltiplica:        ;-- caso della moltiplicazione
44     MUL N2         ;In AL c'è già N1 e dopo la moltiplicazione il risultato è in AX
45     MOV Ris,AX     ;che viene portato in Ris
46     JMP Fine_Se    ;infine si salta all'etichetta Fine_se
47
48 Dividi:           ;-- caso della divisione
49     MOV AH,0       ;Azzerava AH per poter dividere AX per N2(8 bit)
50     DIV N2         ;Dopo la divisione in AL c'è il quoz. e in AH il resto
51     MOV Ris,AX     ;che vengono salvati in Ris parte bassa e alta rispettivamente
52                 ;infine va automaticamente a Fine_se
53 Fine_Se:
54     MOV AH,4Ch     ;Restituzione del controllo al sistema operativo
55     INT 21h
56
57     END Inizio

```