

**Traccia:**

Scrivere un programma in linguaggio assembler 8088 che, presi due dati a e b in memoria, calcola l'espressione  $(a+3)*b$  ponendo il risultato nel registro accumulatore.

```
.MODEL small

.STACK 100h

.DATA
a DW 10 ; Dato 'a' (2 byte)
b DW 5  ; Dato 'b' (2 byte)

.CODE
MAIN PROC

    MOV AX, @DATA ; Inizializza DS al segmento dei dati

    MOV DS, AX

    ; Carica il valore di 'a' in AX
    MOV AX, [a]

    ; Aggiunge 3 a AX
    ADD AX, 3

    ; Carica il valore di 'b' in BX
    MOV BX, [b]

    ; Moltiplica AX per BX
    MUL BX

    ; Il risultato è ora in AX (registro accumulatore)

    ; Puoi utilizzare AX come desideri

    ; Fine del programma
    MOV AH, 4Ch

    INT 21h

MAIN ENDP

END MAIN
```

`.MODEL small:`

Il modello "small" è uno dei modelli di memoria standard nell'assemblaggio x86 e indica che il programma verrà eseguito in un segmento di codice, un segmento di dati e un segmento di stack tutti singoli, ovvero tutti con dimensioni limitate a 64KB. Questo modello è adatto per programmi più semplici e di dimensioni ridotte, dove i dati e il codice non superano la dimensione massima di 64KB.

`.STACK 100h:`

La direttiva `.STACK` imposta la dimensione dello stack (pila) del programma, ovvero la memoria utilizzata per la gestione delle subroutine, delle variabili locali e dei ritorni delle chiamate di funzione.

Nel programma, lo `.STACK 100h` imposta la dimensione dello stack a 256 byte (100h in esadecimale è uguale a 256 in decimale).

Riassumendo, queste direttive iniziali nel codice assembler 8088 definiscono il modello di memoria come "small" e impostano la dimensione dello stack a 256 byte.