

section .data

a dw 5 ; Dichiarazione della variabile a e inizializzazione a 5

b dw 10 ; Dichiarazione della variabile b e inizializzazione a 10

section .text

org 0x100 ; Imposta l'origine del programma a 0x100

start:

mov ax, [a] ; Carica il valore di a nel registro AX

add ax, 3 ; Aggiunge 3 al valore di AX (AX = a + 3)

mov bx, [b] ; Carica il valore di b nel registro BX

mul bx ; Moltiplica AX per BX (AX = (a + 3) \* b)

; Il risultato è ora in AX

; Fine del programma (in un ambiente reale potresti voler terminare con un'istruzione di interruzione o simile)

int 20h ; Termina il programma (DOS interrupt per terminare il programma)

## Spiegazione dettagliata delle istruzioni

### 1. Sezione .data

- o a dw 5: Dichiarazione della variabile a e la inizializza con il valore 5.
- o b dw 10: Dichiarazione della variabile b e la inizializza con il valore 10.

### 2. Sezione .text

- o org 0x100: Imposta l'origine del programma a 0x100, che è l'indirizzo standard per i programmi COM in DOS.

### 3. Start del programma

- o mov ax, [a]: Carica il valore della variabile a nel registro AX.
- o add ax, 3: Aggiunge 3 al valore contenuto in AX, quindi AX ora contiene a + 3.
- o mov bx, [b]: Carica il valore della variabile b nel registro BX.
- o mul bx: Moltiplica il valore di AX per BX, memorizzando il risultato in AX. La moltiplicazione non segna overflow su 16-bit su 8088, quindi il risultato si trova completamente in AX.

### 4. Fine del programma

- o int 20h: Termina il programma utilizzando l'interrupt DOS per uscire.

Quindi, il risultato dell'espressione  $(a+3) \times b$  è ora memorizzato nel registro accumulatore AX