

## Pratica W22D1(2)

### Traccia:



**Esercizio**  
Linguaggio Assembly

#### 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.

### Soluzione:

```
section .data
```

```
    a dw 0      ; variabile a (word, 16 bit)
```

```
    b dw 0      ; variabile b (word, 16 bit)
```

```
    result dw 0 ; risultato (word, 16 bit)
```

```
section .text
```

```
    global _start
```

```
_start:
```

```
    mov ax, [a]    ; carica il valore di a in AX
```

```
    add ax, 3      ; aggiunge 3 a AX
```

```
    mul word [b]   ; moltiplica AX (a+3) per B
```

```
    mov [result], ax ; memorizza il risultato in 'result'
```

```
    ; Esci dal programma
```

```
    mov ah, 4Ch    ; codice di uscita del programma
```

```
    int 21h        ; chiama l'interrupt per terminare il programma
```

## Spiegazione:

### 1. **.data** section:

- Definiamo tre variabili:
  - **a**: Una variabile a 16 bit (word) che conterrà il valore di **a**.
  - **b**: Una variabile a 16 bit (word) che conterrà il valore di **b**.
  - **result**: Una variabile a 16 bit (word) che conterrà il risultato dell'espressione **(a+3)\*b**.

### 2. **.text** section:

- **global \_start**: Indica che **\_start** è il punto di ingresso del programma.

### 3. **\_start**:

- **mov ax, [a]**: Carica il valore contenuto nella variabile **a** in AX. AX è un registro a 16 bit che useremo per effettuare le operazioni.
- **add ax, 3**: Aggiunge 3 al valore contenuto in AX. In questo modo otteniamo il risultato di **a + 3**.
- **mul word [b]**: Moltiplica il valore contenuto nella variabile **b** per il contenuto di AX (che ora è **a + 3**). Il risultato della moltiplicazione viene memorizzato in AX.
- **mov [result], ax**: Memorizza il risultato (contenuto in AX) nella variabile **result**.

### 4. Terminazione del programma:

- **mov ah, 4Ch**: Imposta AH a 4Ch, che è il codice per l'interrupt di terminazione del programma.
- **int 21h**: Chiama l'interrupt 21h, che termina il programma.

In breve, il programma prende il valore di **a**, vi aggiunge 3, quindi moltiplica il risultato per il valore di **b** e memorizza il risultato nella variabile **result**. Infine, il programma termina correttamente.