

# PROGRAMMI ASSEMBLY

## *PROGRAMMA 1 - Somma*

```
; il programma somma due numeri preventivamente inseriti in due registri
codice SEGMENT
    ASSUME CS: codice
inizio:
    mov AH, 03      ;carica il primo numero in AH
    mov AL, 04      ;carica secondo numero in AL
    add AH, AL      ;somma i due numeri, risultato in AH
    mov AH, 4Ch     ;funzione per ritorno al DOS
    int 21h
codice ENDS        ;fine del segmento di codice
END inizio         ;fine del programma
```

## *PROGRAMMA 1 – Somma versione 1*

```
; il programma somma due numeri preventivamente inseriti in due registri
; e mostra il risultato (1 cifra)
codice SEGMENT
    ASSUME CS: codice
inizio:
    mov AH, 03      ;carica il primo numero in AH
    mov AL, 04      ;carica secondo numero in AL
    add AL, AH      ;somma i due numeri, risultato in AL
    add al,48
    mov DL,AL
    mov AH, 02h     ;visualizza il risultato
    int 21h
    mov AH, 4Ch     ;funzione per ritorno al DOS
    int 21h
codice ENDS        ;fine del segmento di codice
END inizio         ;fine del programma
```

## *PROGRAMMA 1 – Somma versione 2*

```
; il programma somma due numeri preventivamente inseriti in due registri
; e mostra il risultato (2 cifre)
codice SEGMENT
    ASSUME CS: codice
inizio:
    mov AH, 19      ; carica il primo numero in AH
    mov AL, 27      ; carica secondo numero in AL
    add AL, AH      ; somma i due numeri, risultato in AL
    mov ah,0        ; calcola decine ed unita'
    mov cl,10
    div cl          ; decine AL - unita' AH
    mov dx,ax       ; sposta il risultato da AX in DX - DL decine - DH unita'
    add dl,48       ; cifra delle decine - ASCII = valore + 48
    mov AH, 02h     ; visualizza il risultato
    int 21h         ; decine
    mov dl,dh       ;
    add dl,48       ;cifra delle unita' - ASCII = valore + 48
    int 21h         ; unita'
    mov AH, 4Ch     ; funzione per ritorno al DOS
    int 21h
codice ENDS        ; fine del segmento di codice
END inizio         ; fine del programma
```

### *PROGRAMMA 2 - Memoria*

```
; il programma pone i due byte 01 e 02 in memoria
dati SEGMENT      ;inizio segmento dati
    num1 DB ?    ;1° locazione con offset num1
    num2 DB ?    ;2° locazione con offset num2
dati ENDS        ;fine segmento dati
```

```
codice SEGMENT
    ASSUME CS:codice, DS:dati
inizio:
    mov AX,dati      ;carica in AX l'ind del segm dati
    mov DS,AX        ;lo copio nel reg del segm. dati
    mov AL,01        ;carica primo numero in AL
    mov num1,AL      ;copia il primo num in memoria
    mov AL,02        ;carica secondo numero in AL
    mov num2,AL      ;copia il secondo num in memoria
    mov AH,4Ch       ;funzione per il ritorno al DOS
    int 21h
codice ENDS        ;fine del segmento di codice
END inizio        ;fine del programma
```

### *PROGRAMMA 3 - Memoria*

```
;il programma preleva tre byte dalla memoria,
;effettua operazioni logiche con essi e pone il risultato in memoria
dati SEGMENT      ;inizio segmento dati
    num1 DB 03h,0CFh,9Dh ;3 byte presenti in memoria
    num2 DB ?      ;locazione con offset num2 per risultato
dati ENDS        ;fine segmento dati
```

```
codice SEGMENT
    ASSUME CS:codice, DS:dati
inizio:
    mov AX,dati      ;carica in AX l'indirizzo del segmento dati
    mov DS,AX        ;lo copia nel registro del segm. dati
    lea SI,num1      ;carica in SI l'offset num1
    lea DI,num2      ;carica in DI l'offset num2
    mov AL,[SI]      ;carica primo numero in AL
    inc SI           ;incrementa il puntatore alla memoria
    mov AH,[SI]      ;carica secondo numero in AH
    and AH,AL        ;effettua AND tra primo e secondo num
    inc SI           ;incrementa il puntatore alla memoria
    mov AL,[SI]      ;carica terzo numero in AL
    or AH,AL         ;effettua con terzo numero
    mov [DI],AH      ;pone il risultato in memoria
    mov AH,4Ch       ;funzione del ritorno al DOS
    int 21h
codice ENDS        ;fine del segmento di codice
END inizio        ;fine del programma
```

## *PROGRAMMA - ASCII*

```
;programma che visualizza la tabella ASCII
codice SEGMENT
    ASSUME CS: codice
inizio:
    mov AH, 02h ; per visualizzare un carattere
    mov DL, 32  ; valore iniziale del ciclo
ciclo:
    int 21h     ; visualizzo il carattere in DL
    inc DL
    cmp DL, 127 ; valore finale del ciclo
    JNE ciclo
    mov AH, 4Ch ; fine programma
    int 21h
codice ENDS
END    inizio
```