

#01 Testo del problema

Dati due byte, scrivere in memoria il maggiore dei due.

#02 Analisi del problema

Per confrontare due numeri è sufficiente farne la sottrazione e verificare due condizioni:

- La nullità del risultato
- Lo stato del flag dei riporti

Se il risultato della sottrazione è nullo, infatti, ovviamente avremo due numeri uguali. Se il risultato è invece diverso da zero, è necessario osservare se è avvenuto un riporto (che nel caso della sottrazione assume il nome di prestito) nell'ultimo bit. Bisogna in pratica guardare se c'è stato qualche overflow. In caso affermativo, il minuendo (primo termine) era minore del sottraendo (secondo termine), in caso negativo il maggiore era invece il minuendo.

A - B	CY	ZF
A > B	0	0
A = B	0	1
A < B	1	0

Dovendo verificare quale dei due numeri in questione sia maggiore, dobbiamo comunque stabilire il nostro comportamento nel caso che essi siano uguali.

Se stabiliamo che in caso di uguaglianza sia considerato "maggiore" il primo, allora possiamo sicuramente evitarci un controllo.

Se il Carry non è settato, siamo infatti sicuri che il minuendo sia maggiore o al limite uguale a sottraendo; in fondo questa situazione ci può andare bene e ci risparmia di controllare il flag di zero.

#03 Progettazione delle routine

Nessuna routine necessaria.

#04 Registri utilizzati

Registri a 8 bit	
AH	▪ Primo numero
AL	▪ Secondo numero
BL	▪ Numero maggiore

Registri a 16 bit	
Nessun registro a 16 bit utilizzato	

#05 Codice del programma

```
p04.asm
;-----
; Direttive all'assemblatore
;-----
DOSSEG                ; Segmentazione DOS
.MODEL  small         ; Modello di memoria
.STACK 100h          ; Dimensione dello stack

;-----
; Data Segment
;-----
.DATA
X      db      100
Y      db      120
MAG    db      ?

;-----
; Code segment
;-----
.CODE

;;;;;;;;;;;;;
; Programma principale
;;;;;;;;;;;;;
Begin: StartupCode      ; Inizio del programma

; Inizializzazione variabili
      MOV  AH , [X]      ; Prelevo il primo numero
      MOV  AL , [Y]      ; Prelevo il secondo numero

; Algoritmo
      CMP  AH , AL      ; Confronto i due numeri
      JC  mY            ; Se salto, è maggiore il secondo
      MOV  BL , AH      ; ...altrimenti è il primo
      JMP  save         ; Fine dell'elaborazione
mY:   MOV  BL , AL      ; Salvo il secondo numero

; Salvataggio risultati
save:  MOV  [MAG] , BL   ; Salvo il numero maggiore

; Operazioni finali
ExitCode 0            ; Esecuzione corretta del programma
END Begin            ; Fine del programma
```

#06 Prove effettuate

Prova 01			
Significato della prova	▪ E' maggiore il secondo numero		
Valori in input	X	=>	100
	Y	=>	120
Valori attesi in output	MAG	=>	120
Valori ottenuti in output	MAG	=>	120

Prova 02			
Significato della prova	▪ E' maggiore il primo numero		
Valori in input	X	=>	120
	Y	=>	100
Valori attesi in output	MAG	=>	120
Valori ottenuti in output	MAG	=>	120

Prova 03			
Significato della prova	▪ I due numeri sono uguali		
Valori in input	X	=>	120
	Y	=>	120
Valori attesi in output	MAG	=>	120
Valori ottenuti in output	MAG	=>	120