

```

1 ;Siano le variabile A e B in memoria che contengono due valori a 8 bit
2 ;Se A<B allora ---> Min=A
3 ; altrimenti ---> Min=B
4 ;=====
5 ; p257_E6a.asm
6 ;=====
7
8 .MODEL SMALL
9
10 .DATA ;direttiva che dichiara il segmento Dati
11 A DB 150
12 B DB 017h
13 Min DB ? ;alla fine del programma conterrà il minimo tra A e B
14
15 .STACK ;direttiva che dichiara il segmento STACK di 1024 byte
16
17 .CODE ;direttiva che dichiara l'inizio del segmento Codice
18
19 Inizio:
20 ;Inizializzazione dei registri di segmento
21 MOV AX,@DATA ;in AX viene posto l'indirizzo del segmento
22 MOV DS,AX ;Dati per poterlo assegnare al registro DS
23
24 ;programma
25 MOV AL,A ;Si assegna al registro AL il valore contenuto nella variabile A
26
27 ;-- Inizio SE
28 CMP AL,B ;Si confronta AL con il contenuto di B
29 JB Allora ; --- Se AL<B ==> si va all'etichetta Allora
30 ;-- caso Altrimenti
31 MOV BL,B ;assegna a BL il valore contenuto in B
32 MOV Min,BL ;e si assegna a Min il valore contenuto in B
33 JMP Fine_Se ;infine si salta all'etichetta Fine_se
34 Allora: ;-- caso Allora
35 MOV Min,AL
36
37 Fine_Se:
38 MOV AH,4Ch ;Restituzione del controllo al sistema operativo
39 INT 21h
40
41 END Inizio

```

```

1 ;Calcolare la somma di due numeri sommando un'unità alla volta
2 ; somma = num1;
3 ; i = 0;
4 ; while (i < num2)
5 ;     { somma++;
6 ;       i++;
7 ;     }
8 ;=====
9 ; p258_E6b.asm
10 ;=====
11 .MODEL SMALL
12
13 .DATA                ;direttiva che dichiara il segmento Dati
14     Num1      DB    254
15     Num2      DB    8
16     Somma     DW    ?    ;alla fine del programma conterrà il risultato della somma
17 .STACK                ;direttiva che dichiara il segmento STACK di 1024 byte
18 |.CODE                ;direttiva che dichiara l'inizio del segmento Codice
19
20 Inizio:
21                ;Inizializzazione dei registri di segmento
22     MOV  AX,@DATA    ;in AX viene posto l'indirizzo del segmento
23     MOV  DS,AX       ;Dati per poterlo assegnare al registro DS
24
25 ;programma
26     MOV  AH,0        ;Si azzera la parte alta di AX
27     MOV  AL,Num1     ;Si assegna a AL il primo numero (1 byte)
28     MOV  Somma,AX    ;Si pone in Somma il contenuto di AX (in realtà Num1)
29     MOV  BL,0        ;Utilizziamo BL come contatore (i=0)
30
31 Inizio_ciclo:
32     CMP  BL,Num2     ;Si confronta il contenuto di BL con Num2
33     JGE  Fine_ciclo  ;Se BL>=Num2 si esce dal ciclo
34     INC  Somma       ;Si incrementa la Somma di 1
35     INC  BL          ;Si incrementa il contatore di 1
36     JMP  Inizio_ciclo ;Si torna su (salto incondizionato)
37
38 Fine_ciclo:
39     MOV  AH,4Ch      ;Restituzione del controllo al sistema operativo
40     INT  21h
41     END  Inizio

```

```

1 ;Dato un numero N dichiarato DW calcolare da quante cifre è composto
2 ; cifre =0;
3 ; do{ N=N/10;
4 ;     cifre++;
5 ;     }while(N >=1);
6 ;=====
7 ; p258_E6c.asm
8 ;=====
9 .MODEL SMALL
10
11 .DATA                ;direttiva che dichiara il segmento Dati
12     N                DW    2786
13     Cifre            DB    ?    ;alla fine del programma conterrà il numero
14                                ; di cifre di cui è composto N
15
16 .STACK                ;direttiva che dichiara il segmento STACK di 1024 byte
17
18 .CODE                ;direttiva che dichiara l'inizio del segmento Codice
19
20 Inizio:
21                                ;Inizializzazione dei registri di segmento
22     MOV    AX,@DATA    ;in AX viene posto l'indirizzo del segmento
23     MOV    DS,AX      ;Dati per poterlo assegnare al registro DS
24
25 ;programma
26     MOV    Cifre,0    ;Si inizializza Cifre con 0
27     MOV    BX,10     ;Si assegna a BX il valore 10 (valore del divisore)
28     MOV    AX,N      ;Si assegna N ad AX prevedendo di fare la divisione poi per BX
29
30 Inizio_ciclo:
31     MOV    DX,0      ;ogni volta azzerò DX perché dopo aver fatto la divisione
32                                ;ci viene scritto il resto
33     DIV    BX        ;Divido DXAX per BX e il quoziente si trova in AX
34                                ;(il resto è in DX)
35     INC    Cifre     ;Incremento il numero delle cifre trovate
36     CMP    AX,1     ;Si confronta AX con il valore 1
37     JGE    Inizio_ciclo ;Se AX>=1 ==> si torna su per dividere
38
39     MOV    AH,4Ch    ;Restituzione del controllo al sistema operativo
40     INT    21h
41     END    Inizio

```