

Esercizio di programmazione

sino a 12 punti – è possibile consultare solamente l'instruction set Intel - tempo: 60 minuti

Il prezzo di un parcheggio è pari a X Euro per ogni periodo di Y minuti. Per eventuali minuti di un periodo non completo sono addebitati comunque X Euro.

Esempio:

X : 1 Euro

Y : 40 minuti

Orario di ingresso: 12.47

Orario di uscita: 18.14

Il tempo di permanenza corrisponde a 8 periodi interi più 7 minuti. Il costo del parcheggio è 9 Euro.

Si scriva una procedura **costoParcheggio** in linguaggio Assembly 8086 in grado di calcolare il costo per il parcheggio. X e Y sono dichiarati come costanti. Gli orari di ingresso e di uscita sono memorizzati ciascuno in un vettore di 2 byte: il primo indica l'ora e il secondo i minuti. La procedura **costoParcheggio** riceve l'offset dei due vettori tramite stack e restituisce il costo del parcheggio attraverso il registro AX.

Si assuma che gli orari siano sempre consecutivi e appartenenti alla stessa giornata.

Di seguito un esempio di programma chiamante:

```
X EQU 1
Y EQU 40

.MODEL small
.STACK
.DATA
ora_in  db 12, 47
ora_out db 18, 14

.CODE
.STARTUP
    LEA AX, ora_in
    PUSH AX
    LEA AX, ora_out
    PUSH AX
    CALL costoParcheggio
    ADD SP, 4
.EXIT
```

Soluzione proposta

```
X EQU 1
Y EQU 40

.MODEL small
.STACK
.DATA
ora_in  db 12, 47
ora_out db 18, 14

.CODE
.STARTUP
    LEA AX, ora_in
    PUSH AX
    LEA AX, ora_out
    PUSH AX
    CALL costoParcheggio
    ADD SP, 4
.EXIT

costoParcheggio proc
    mov bp, sp
    mov si, [bp+4]
    mov di, [bp+2]
    mov al, [di]
    mov ch, 60
    mul ch
    add al, [di+1]
    adc ah, 0
    mov dx, ax
    mov al, [si]
    mul ch
    add al, [si+1]
    adc ah, 0
    sub dx, ax
    mov ax, dx
    mov ch, Y ; minuti
    div ch
    cmp ah, 0
    jz next
    add al, 1
next: mov ch, X ; X euro ogni Y minuti
    mul ch ; valore di ritorno in AX
    ret
costoParcheggio endp

end
```