Problema: calcolo del giorno dell'anno

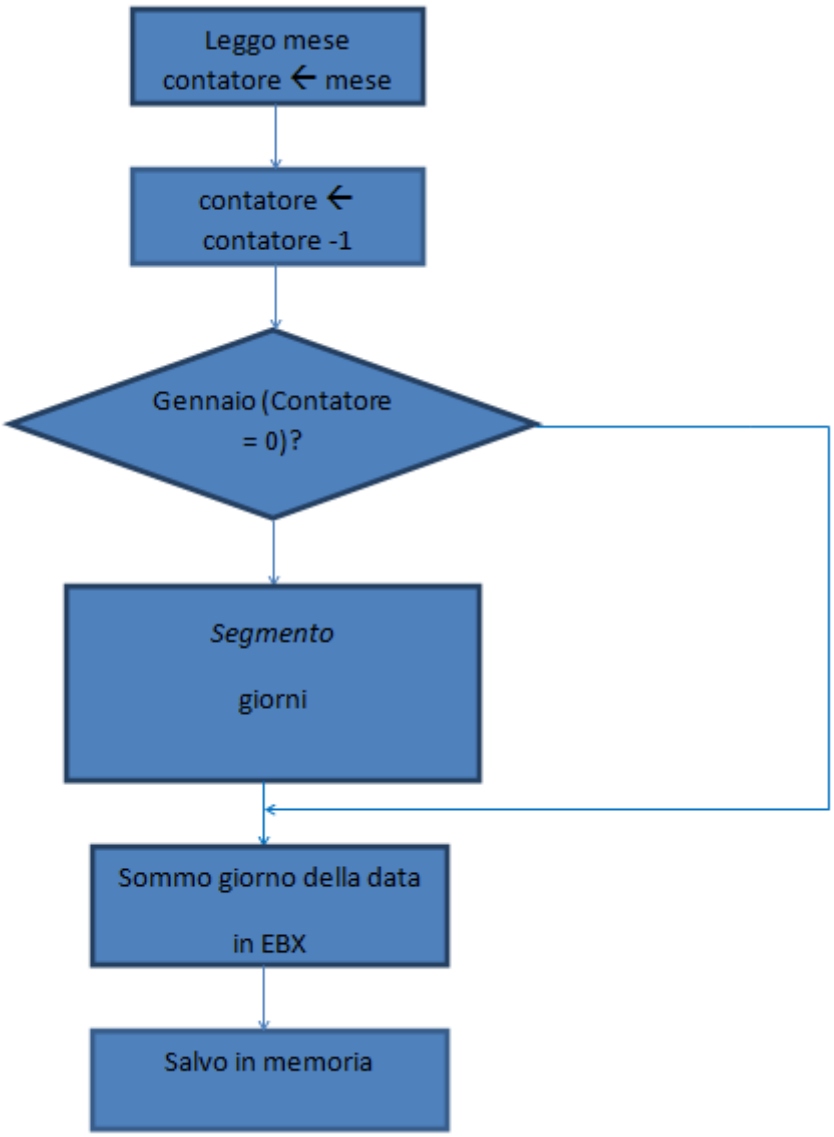
Dato giorno e mese nell'anno, proporre, in **ambiente Jasmin**, un programma che calcoli il numero progressivo del giorno nell'anno (da 1 a 365)

INPUT
il giorno si trova in memoria all'indirizzo **0x30** cioè dopo le 12x4 celle che contengono i giorni di ogni mese

il mese si trova in memoria all'indirizzo **0x34**

OUTPUT
il progressivo calcolato va scritto all'indirizzo 0x48

Flusso



PROCEDIMENTO
somma giorni in EBX
contatore mesi in EAX
punto in memoria con EDX

← *poi miglioramento con uso registro C sfruttando istruzione LOOP*

data:
dd 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 ***; non bisestile***
dd 3,3 ; ***3 marzo*** *da modificare* ← da programma
; leggo il mese dalla memoria
; decremento **eax contatore** mesi
; …. poi useremo il registro C
; sfruttando istruzione LOOP
; se sono in gennaio non faccio iterazione

giorni:
;sommo i giorni del mese corrente
;incremento il puntatore
;decremento il contatore
; se non zero salto a giorni

meseCorrente:
;aggiungo il giorno della data richiesta
;scrivo il risultato in memoria

Attività:

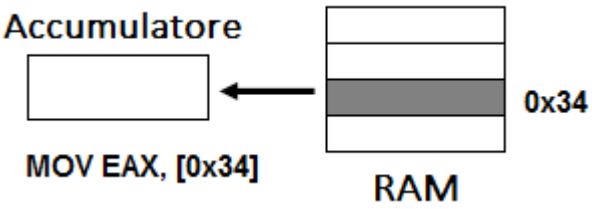
1) proporre il codice assembly come da commento
nb: con *leggere da memoria* si intende l'operazione illustrata

2) modificare il codice con uso di registro C sfruttando istruzione LOOP *(si veda l'esempio)*

Memory

desc | hex | highlight

address	signed int
0x0	31
0x4	28
0x8	31
0xC	30
0x10	31
0x14	30
0x18	31
0x1C	31
0x20	30
0x24	31
0x28	30
0x2C	31
0x30	3
0x34	3



Esempio : Registro C usato come contatore

```
; uso istruzione LOOP
data:
dd 10
MOV EBX, data          ; puntatore
MOV ECX, [data]        ; contatore registro C
salta:
  MOV AL, [EBX]
  INC EBX
                       ; l'istruzione LOOP: decrementa CX e, se CX non è 0, salta all'etichetta specificata
  LOOP salta
```

Dall'URL
http://wwwi10.lrr.in.tum.de/~jasmin/commands.html

JZ
LOOP

LOOP-Loop while CX is not zero

Usage: LOOP label

Arguments: label: label that points to the beginning of the iteration

Effects: Decrements CX (not modifying any flags) and jumps to label if CX is not zero

Flags: none

Misc: Create a loop of N iterations by copying N to CX (MOV CX,N), adding a label to the beginning of the block of operations to be repeated and a LOOP statement to its end.

LOOPE
LOOPNE
LOOPNZ
LOOPZ
MOV

Set dell'8088/86

LOOP gira mentre CX non è zero
LOOPE gira mentre è uguale
LOOPNE gira mentre non è uguale
LOOPZ gira mentre è zero
LOOPNZ gira mentre non è zero

LOOP - Salta incondizionatamente CX volte.
LOOPD - Salta incondizionatamente ECX volte.

se **ECX**=00000000 *prosegue con l'istruzione successiva*.

se **ECX** è diverso da 00000000 ***salta*** all'indirizzo suggerito dall'etichetta

non appartiene al Set dell'8088/86; si usa solo con **80386/486**.

LOOPE - Salta CX volte se uguale

LOOPED - Salta ECX volte se uguale

LOOPEW - Salta CX volte se uguale

LOOPNED - Salta ECX volte se non è uguale

LOOPNEW - Salta CX volte se non è uguale

LOOPNZD - Salta ECX volte se non è zero

LOOPNZW - Salta CX volte se non è zero

LOOPW - Salta incondizionatamente CX volte

LOOPZD - Salta ECX volte se è zero

LOOPZW - Salta CX volte se è zero

non appartengono al Set dell'8088/86; si usa solo con **80386/486**.

LOOPNE - Salta CX volte se non uguale.

LOOPNZ - Salta CX volte se non è zero.

LOOPZ - Salta CX volte se zero