

13_01 Primo appello invernale

Scrivere un sottoprogramma nel linguaggio assembly AVR, per un microcontrollore della famiglia XMEGA, in grado di individuare il numero di occorrenze di una stringa di 3 caratteri all'interno di una stringa di 16 caratteri. Le due stringhe sono codificate secondo la codifica ASCII. Le due stringhe sono collocate nella memoria dati estesa, con i caratteri in posizioni consecutive a partire da un indirizzo base il cui valore è contenuto nel puntatore X per la stringa da 16 caratteri e nel puntatore Y per la stringa da cercare da 3 caratteri. Il risultato deve essere lasciato nel registro R0. Ovviamente, il sottoprogramma non deve alterare nessun altro registro di lavoro.

```
/* Nota: l'esercizio si risolve annidando due subroutine, una che rivela l'uguaglianza tra due stringhe di 3 caratteri, lasciando C vero se riscontra l'uguaglianza, e una che scorre per 14 volte la stringa lunga, cercando la presenza della stringa piccola. Un piccolo tranello può essere rappresentato dalla presenza di stringhe interallacciate: in "aaaaaaaa0aaaaaaaa0" la stringa "aaa" è contenuta 10 volte e non 4!
```

```
*/  
compare3:          //confronta le stringhe di 3 byte puntate da X e Y  
    push R16  
    push R17  
    push R18  
    push R19  
    clr R16         //resta a 0 se c'è uguaglianza  
    ldi R19,3       //esegue 3 confronti  
    cp1:  
    ld R17,X+  
    ld R18,Y+  
    cpse R17,R18  
    inc R16  
    dec R19  
    brne cp1  
    sbiw XH:XL,2    //il puntatore X punterà al carattere successivo  
    sbiw YH:YL,3    //il puntatore Y viene ripristinato  
    clc  
    tst R16         //controlla se stringhe uguali  
    brne cp2  
    sec            //se uguali mette a 1 il carry  
    cp2:  
    pop R19  
    pop R18  
    pop R17  
    pop R16  
ret  
  
search:           //subroutine richiesta  
    push R1  
    push R16  
    clr R0         //inizializza il risultato  
    clr R1  
    ldi R16,14
```

```
loop:
    rcall compare3    //cerca la stringa da 3 caratteri in quella da 16
    adc R0,R1         //incrementa R0 se trovata
    dec R16
    brne loop
    sbiw XH:XL,14     //ripristina il puntatore X
    pop R16
    pop R1
ret
```