

Esercizio 1:

Assembler

Sia array la seguente sequenza di 16 byte: 0x1A, 0x47, 0x34, 0xC5, 0x9B, 0x02, 0x6D, 0x8E, 0x9B, 0x1D, 0x47, 0x60, 0x29, 0x3A, 0x9B, 0x11. Scrivere un programma Assembler che permette, interattivamente, di cercare occorrenze di specifici byte all'interno di questa sequenza.

Il programma, ciclicamente, attende un comando valido e lo esegue. Ignora qualunque carattere e comando non valido, rimanendo in attesa.

I comandi validi sono:

- s: Avvia una nuova ricerca. Ha come argomento il byte da cercare, per esempio s 1A.
- n: Continua la ricerca in corso, ripartendo da *dopo* l'ultima occorrenza trovata. Il comando è non valido se non c'è una ricerca in corso.
- f: Termina il programma.

I comandi s e n cercano la prima occorrenza del byte.

Quando viene trovata un'occorrenza, e non è stata terminata la ricerca, stampano un rapporto nel seguente formato:

```
Trovata occorrenza di  
1A  
Conteggio attuale :  
1
```

Quando la ricerca giunge al termine, stampano un rapporto nel seguente formato:

```
Scansione array terminata .  
Totale :  
1  
occorrenze di  
1A
```

Note

- Ciascuna chiamata del comando `s` e `n` esegue una sola ricerca per la prossima occorrenza del byte, per poi tornare in attesa di un comando.
- Il comando `s` comincia la nuova ricerca dal primo elemento dell'array.
- È possibile avviare una nuova ricerca senza terminare una precedente.
- All'inserimento del comando `s`, il programma stampa uno spazio e comincia la lettura dell'argomento.

- Per la lettura di byte è sufficiente la validazione fatta da inbyte.
- Si faccia uso, dove si ritiene applicabile, delle istruzioni stringa.

Una dichiarazione dell'array in codice Assembler ed un esempio di output sono allegati al link:

<https://tinyurl.com/4ts65hzc>

Si ponga attenzione alla formattazione dell'esempio di output, che fa parte delle specifiche.