

Esercizio 1: Assembler

Scrivere un programma Assembler che si comporta come segue.

1. Legge da tastiera due stringhe esadecimali di 8 e 4 cifre, che interpreta come rappresentazioni binarie di due numeri interi a , rappresentabile su 32 bit, e b , rappresentabile su 16 bit.
2. Calcola, se esistono, gli interi q ed r , rappresentabili su 16 bit e tali per cui

$$\begin{cases} a = q \cdot b + r, \\ ABS(r) < ABS(b), \\ sgn(r) = sgn(a). \end{cases}$$

3. Se q ed r esistono, ne stampa su linee distinte la rappresentazione binaria in notazione esadecimale. Altrimenti, stampa "NO DIV".
4. Termina.

Note

- Per la lettura dei numeri interi, sono sufficienti i controlli fatti da `inlong/inword/inbyte`.
- Per la stampa dei numeri interi, è sufficiente quanto fatto da `outlong/outword/outbyte`.
- Non è permesso l'uso di istruzioni `DIV` o `IDIV`.

Esempi di output sono allegati al link: <https://tinyurl.com/a4h442nb>

Si ponga attenzione alla formattazione degli output, che fa parte delle specifiche.

La documentazione Assembler in formato PDF è scaricabile al link: <https://tinyurl.com/ys9euyc9>