

Esercizio 1: Assembler

Scrivere un programma Assembler che realizza una calcolatrice per interi, come segue:

- Legge e fa eco di un numero intero $x \in [-99, +99]$.
- Legge e fa eco di un operatore tra +, -, * o /.
- Legge e fa eco di un numero intero $y \in [-99, +99]$.
- Calcola e stampa su una nuova riga il risultato dell'operazione, o "ERR" se questa è impossibile da calcolare.
- Se l'operazione è impossibile, termina. Altrimenti, torna al punto 1 chiedendo una nuova operazione.

Note:

- I numeri dei numeri interi vanno inseriti come segno e modulo. Per la lettura del segno (+ o -) il programma deve ignorare caratteri inattesi. Per lettura del modulo, è sufficiente la validazione fatta da `indecim1`.
- Per il valore 0, l'inserimento di + o - è indifferente.
- Nel caso dell'operatore divisione (/), è di interesse solo il quoziente.
- Si ricorda che l'istruzione CBW estende un intero in `%al` su `%ax`, mentre CWDE fa lo stesso da `%ax` ad `%eax`.

Un esempio di output è allegato in formato .txt al link: <https://tinyurl.com/msmppxrk>
Si ponga attenzione alla formattazione di questo file, che fa parte delle specifiche. Errori di formattazione non saranno presi in considerazione durante la valutazione, purché siano risolti in autocorrezione.