

Simulazione verifica di Laboratorio: Assembly x86 (parte relativa alla programmazione)

Obiettivo: Realizzare un programma per la "pulizia" e normalizzazione di un flusso di dati in ingresso.

Traccia dell'Esercizio

Sviluppa un programma in linguaggio Assembly (per architettura 8086) che gestisca l'elaborazione di una sequenza di caratteri secondo le seguenti specifiche:

1. Ricezione Dati:

- Il programma deve richiedere all'utente l'inserimento di **7 caratteri**.
- Ogni carattere deve essere analizzato in tempo reale prima del salvataggio.

2. Logica di Elaborazione:

- **Numeri:** Se l'utente inserisce una cifra numerica (0-9), questa non deve essere memorizzata; al suo posto, salva nell'array il carattere trattino (-).
- **Lettere Minuscole:** Se l'utente inserisce una lettera minuscola (a-z), il programma deve convertirla in maiuscola prima di salvarla.
- **Altri Caratteri:** Qualsiasi altro simbolo o lettera maiuscola va salvato senza modifiche.

3. Gestione Memoria:

- I 7 caratteri risultanti dall'elaborazione devono essere memorizzati in un array chiamato BUFFER definito nel segmento dati.
- Per il riempimento dell'array è obbligatorio l'uso del registro **SI** come puntatore.

4. Output:

- Al termine della fase di inserimento, il programma deve andare a capo e stampare il messaggio: "CODICE PULITO: ".
- Subito dopo, deve mostrare il contenuto dell'array BUFFER.

Vincoli Tecnici e Valutazione

Per ottenere il punteggio massimo, il codice dovrà rispettare i seguenti criteri:

- **Modularità:** Utilizzo corretto del ciclo LOOP per la fase di input.
- **Integrità del Contatore:** Gestione dello **Stack** per preservare il registro CX durante le chiamate di sistema INT 21H.

- **Efficienza:** Uso corretto dei salti condizionati (CMP e istruzioni di tipo Jxx) per implementare la logica di filtraggio senza ripetizioni di codice inutili.
- **Pulizia:** Commentare le sezioni principali del codice (Input, Logica, Output).

Nota per lo studente

Ricorda che la trasformazione da minuscolo a maiuscolo in ASCII si ottiene manipolando il valore numerico del carattere. Presta attenzione a non "sovrapporre" le condizioni di filtro: un carattere trasformato in trattino non deve essere successivamente processato dal filtro delle maiuscole.