```
<!DOCTYPE HTML>

<html>
    <head>
        <!-- Cucchi Francesco 4BI comandi.html -->
        <title>Comandi</title>
        <link rel="stylesheet" type="text/css" href="..\..\CSS\style.css">
    </head>

    <body>
        <nav class="flexbox">
            <a href="..\..\index.html"> HOME </a>
            <a href="assembly.html">ASSEMBLY</a>
            <span>COMANDI</span>
            <a href="interrupt.html">INTERRUPT</a>
        </nav>
        <h1>COMANDI</h1>
        <ul>
            <li>
                ASSEGNAZIONE:
                <ul>
                    <li>
                        MOV (Move):
                        <ul>
                            <li>Sintassi: MOV dest, src</li>
                            <li>Descrizione: Copia il valore di src (sorgente) in dest (destinazione).</li>
                            <li>Esempio: MOV AX, 10 (assegna il valore 10 al registro AX)</li>
                        </ul>
                    </li>
                    <li>
                        LEA (Load Effective Address):
                        <ul>
                            <li>Sintassi: LEA dest, src</li>
                            <li>Descrizione: Carica l'indirizzo effettivo di src in dest. Questo &egrave; utile per indirizzare la memoria in modo indiretto.</li>
                            <li>Esempio: LEA BX, [ARRAY] (carica l'indirizzo di base dell'array ARRAY nel registro BX)</li>
                        </ul>
                    </li>
                    <li>
                        PUSH:
                        <ul>
                            <li>Sintassi: PUSH src</li>
                            <li>Descrizione: Mette il valore di src sullo stack.</li>
                            <li>Esempio: PUSH AX (mette il valore del registro AX sullo stack)</li>
                        </ul>
                    </li>
                    <li>
                        POP:
                        <ul>
                            <li>Sintassi: POP dest</li>
                            <li>Descrizione: Rimuove il valore in cima allo stack e lo copia in dest.</li>
                            <li>Esempio: POP BX (rimuove il valore in cima allo stack e lo copia nel registro BX)</li>
                        </ul>
                    </li>
                    <li>
                        XCHG:
                        <ul>
                            <li>Sintassi: XCHG dest, src</li>
                            <li>Descrizione: Scambia i valori di dest e src.</li>
                            <li>Esempio: XCHG AX, BX (scambia i valori dei registri AX e
```

```
                        BX)</li>
                    </ul>
                </li>
            </ul>
        </li>
        <li>
            CONFORNTO/SALTO:
            <ul>
                <li>
                    CMP (Compare):
                    <ul>
                        <li>Sintassi: CMP op1, op2</li>
                        <li>Descrizione: Sottrae op2 da op1 e imposta i flag di
                        stato del processore in base al risultato. Non modifica i
                        valori di op1 o op2.</li>
                        <li>Esempio: CMP AX, BX (confronta il valore del registro AX
                        con quello del registro BX)</li>
                    </ul>
                </li>
                <li>
                    JMP (Jump):
                    <ul>
                        <li>Sintassi: JMP label</li>
                        <li>Descrizione: Salta incondizionatamente all'indirizzo di
                        memoria specificato da label.</li>
                        <li>Esempio: JMP ESEMPIO (salta all'indirizzo di memoria
                        etichettato come ESEMPIO)</li>
                    </ul>
                </li>
                <li>
                    Jxx (Jump if xx):
                    <ul>
                        <li>Sintassi: Jxx target</li>
                        <li>Descrizione: Salta all'indirizzo di memoria specificato
                        da target se la condizione specificata dal codice xx &egrave;
                         vera.</li>
                        <li>Esempi: JL (Jump if Less) e JG (Jump if Greater),
                        eseguono il salto solo se il primo operando &egrave;
                        maggiore/minore del secondo</li>
                    </ul>
                </li>
            </ul>
        </li>
        <li>
             STRUTTURA IF:
             <ol>
                <li>Confronto: Utilizzare il comando CMP per confrontare due valori.
                Questo imposta i flag di stato del processore in base al risultato
                del confronto.</li>
                <li>Salto condizionale: Utilizzare un comando Jxx (Jump Conditional)
                per saltare a un'altra parte del codice se la condizione specificata
                &egrave; vera.</li>
                <li>Blocco di codice "if": Il codice che verr&agrave; eseguito se la
                condizione &egrave; vera va inserito dopo il salto condizionale.</li>
                <li>Blocco di codice (opzionale) "else": Il codice che verr&agrave;
                eseguito se la condizione &egrave; falsa va inserito dopo il salto
                incondizionale.</li>
             </ol>
        </li>
        <li>
            STRUTTURA CICLO PRE-CONDIZIONALE:
            <ol>
                <li>Inizializzazione: Imposta il valore iniziale della variabile di
                controllo del ciclo.</li>
                <li>Condizione: Verifica se la condizione del ciclo &egrave; vera.
                </li>
                <li>Corpo del ciclo: Esegui il codice all'interno del ciclo se la
```

```
                    condizione &egrave; vera.</li>
                    <li>Aggiornamento: Aggiorna la variabile di controllo del ciclo.</li>
                    <li>Salto: Salta indietro all'inizio del ciclo per verificare
                    nuovamente la condizione.
                    </li>
                </ol>
            </li>
            <li>
                STRUTTURA CICLO POST-CONDIZIONALE:
                <ol>
                    <li>Corpo del ciclo: Esegui il codice all'interno del ciclo.</li>
                    <li>Condizione: Verifica se la condizione del ciclo &egrave; vera.
                    </li>
                    <li>Salto: Salta indietro all'inizio del ciclo se la condizione
                    &egrave; vera.</li>
                </ol>
            </li>
        </ul>
    </body>
</html>
```