# Analisi del Codice Assembly

In questa analisi, esamineremo un esempio di codice Assembly per la CPU x86. Il codice fornito esegue una serie di istruzioni di base che coinvolgono l'uso di registri e operazioni aritmetiche. Analizzeremo ogni istruzione singolarmente, descrivendo il suo scopo e la sua funzione nel contesto del programma. Utilizzeremo la conversione da esadecimale a decimale per facilitare la comprensione dei valori utilizzati nelle istruzioni.

0x00001141 <+8>: mov EAX, 0x20

**Carica il valore 32 nel registro EAX.**

0x00001148 <+15>: mov EDX, 0x38

**Carica il valore 56 nel registro EDX.**

0x00001155 <+28>: add EAX, EDX

**Aggiunge il valore contenuto nel registro EDX al valore contenuto nel registro EAX..**

0x00001157 <+30>: mov EBP, EAX

**Copia il valore attuale del registro EAX nel registro EBP.**

**0x0000115a** <+33>: cmp EBP, 0xa

**Confronta il valore nel registro EBP con il valore esadecimale 0xa (10 in decimale).**

0x0000115e <+37>: jge 0x1176 <main+61>

**Salta all'indirizzo 0x1176 (etichetta <main+61>) se il valore nel registro EBP è maggiore o uguale a 0xa.**

0x0000116a <+49>: mov eax, 0x0

**Carica il valore 0x0 (0 in decimale) nel registro eax.**

0x0000116f <+54>: call 0x1030 <printf@plt>

**Chiama la funzione printf situata all'indirizzo 0x1030.**