

# Report Esercizio

W22-D2



---

**Redatto da Andrea Sciattella**

22/07/2024

**TRACCIA**

---

Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice.

Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

- 0x00001141 <+8>: mov EAX,0x20
- 0x00001148 <+15>: mov EDX,0x38
- 0x00001155 <+28>: add EAX,EDX
- 0x00001157 <+30>: mov EBP, EAX
- 0x0000115a <+33>: cmp EBP,0xa
- 0x0000115e <+37>: jge 0x1176 <main+61>
- 0x0000116a <+49>: mov eax,0x0
- 0x0000116f <+54>: call 0x1030 <printf@plt>

**SVOLGIMENTO ESERCIZIO**

---

1. 0x00001141 <+8>: mov EAX,0x20: *(Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX).*
2. 0x00001148 <+15>: mov EDX,0x38: *(Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX).*
3. 0x00001155 <+28>: add EAX,EDX: *(Somma il valore nel registro EDX (56) al valore nel registro EAX (32), il risultato (88) viene salvato nel registro EAX).*
4. 0x00001157 <+30>: mov EBP, EAX: *(Copia il valore del registro EAX (88) nel registro EBP).*
5. 0x0000115a <+33>: cmp EBP,0xa: *(Confronta il valore del registro EBP (88) con il valore esadecimale 0xa (10 in decimale). Questa istruzione imposta i flag di condizione basati sulla differenza tra i due valori).*
6. 0x0000115e <+37>: jge 0x1176 <main+61>: *(Salta all'indirizzo 0x1176 se il risultato del confronto precedente (cmp) indica che EBP è maggiore o uguale a 0xa (10). In questo caso, il valore di EBP (88) è maggiore di 10, quindi il salto verrà eseguito).*
7. 0x0000116a <+49>: mov eax,0x0: *(Questa istruzione verrà eseguita solo se il salto non viene effettuato, il che non è il caso qui) Carica il valore 0x0 (0 in decimale) nel registro EAX.*
8. 0x0000116f <+54>: call 0x1030 <printf@plt>: *(Chiama la funzione printf situata all'indirizzo 0x1030. Questa chiamata di funzione potrebbe stampare un messaggio a schermo, ma il comportamento esatto dipende dai parametri e dalla configurazione della funzione printf).*