

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

1. Assegna il valore 32 in esadecimale al registro EAX
2. Assegna il valore 56 in esadecimale al registro EDX
3. Somma il valore tra i due registri e lo assegna al registro EAX
4. Assegna il valore contenuto nel registro EAX al registro EBP
5. Confronta il valore 10 in esadecimale col valore contenuto nel registro EBP (88), settando i flag **ZF** e **CF**. Nel caso specifico:
 - **Zero flag (ZF):** Verrà impostato a 0 perché 88 non è uguale a 10.
 - **Carry flag (CF):** Verrà impostato a 0 perché non c'è riporto durante la sottrazione (88 - 10 non richiede operazioni sulle decine).
6. Verifica il valore dei flag (ZF,CF) per stabilire se effettuare un salto alla istruzione 0x1176, mentre <main+61> è un modo per identificare l'istruzione di destinazione in modo più leggibile.
7. Assegna il valore 0 in esadecimale al registro EAX
8. Chiamata alla funzione print per stampare qualcosa