

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>
```

Spiegazione:

1. **mov EAX, 0x20:** Questa istruzione sposta il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
2. **mov EDX, 0x38:** Questa istruzione sposta il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
3. **add EAX, EDX:** Questa istruzione aggiunge il contenuto del registro EDX al registro EAX e memorizza il risultato in EAX. In altre parole, aggiunge 56 a 32 (valore corrente di EAX).
4. **mov EBP, EAX:** Questa istruzione copia il valore corrente del registro EAX nel registro EBP.
5. **cmp EBP, 0xa:** Questa istruzione compara il valore nel registro EBP con il valore esadecimale 0xa (10 in decimale).
6. **jge 0x1176 <main+61>:** Questa istruzione salta all'indirizzo di memoria 0x1176 se il risultato dell'ultima operazione di confronto è maggiore o uguale. Altrimenti, prosegue con l'esecuzione sequenziale.
7. **mov EAX, 0x0:** Questa istruzione assegna il valore esadecimale 0x0 al registro EAX.
8. **call 0x1030 <printf@plt>:** Questa istruzione chiama la funzione printf, presumibilmente per stampare un messaggio a schermo utilizzando il valore 0x0 come argomento.