

Assembly x86

Traccia:

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.

```
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>
```

Soluzione:

1. **mov EAX,0x20**: Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
2. **mov EDX,0x38**: Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
3. **add EAX,EDX**: Aggiunge il contenuto del registro EDX al registro EAX e memorizza il risultato in EAX.
4. **mov EBP,EAX**: Copia il contenuto del registro EAX nel registro EBP.
5. **cmp EBP,0xa**: Confronta il contenuto del registro EBP con il valore decimale 10 (0xa in esadecimale).
6. **jge 0x1176 <main+61>**: Salta all'indirizzo 0x1176 (etichettato come main+61) se il risultato dell'ultima comparazione è maggiore o uguale (jge = jump if greater than or equal).
7. **mov eax,0x0**: Carica il valore esadecimale 0x0 (0 in decimale) nel registro EAX.
8. **call 0x1030 <printf@plt>**: Chiama la funzione printf situata all'indirizzo 0x1030.