

```

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>

```

- 1) **0x00001141 <+8>: mov EAX,0x20**
 - a. Sposta il valore immediato 0x20 (32 in decimale) nel registro EAX.
- 2) **0x00001148 <+15>: mov EDX,0x38**
 - a. Sposta il valore immediato 0x38 (56 in decimale) nel registro EDX.
- 3) **0x00001155 <+28>: add EAX,EDX**
 - a. Somma il valore nel registro EDX (56) al valore nel registro EAX (32), e memorizza il risultato in EAX. Dopo questa operazione, EAX sarà 88.
- 4) **0x00001157 <+30>: mov EBP, EAX**
 - a. Sposta il valore nel registro EAX (88) nel registro EBP.
- 5) **0x0000115a <+33>: cmp EBP,0xa**
 - a. Confronta il valore nel registro EBP (88) con il valore immediato 0xa (10 in decimale). Non modifica nessun registro, ma imposta i flag di condizione in base al risultato della comparazione.
- 6) **0x0000115e <+37>: jge 0x1176 <main+61>**
 - a. Salta all'indirizzo 0x1176 se il valore nel registro EBP è maggiore o uguale a 10. In questo caso, il salto avverrà perché 88 è maggiore di 10.
- 7) **0x0000116a <+49>: mov eax,0x0**
 - a. Sposta il valore immediato 0x0 (0 in decimale) nel registro EAX.
- 8) **0x0000116f <+54>: call 0x1030 <printf@plt>**
 - a. Chiama la funzione printf utilizzando l'indirizzo nella Procedure Linkage Table (PLT) che è 0x1030.

Questa sequenza di istruzioni esegue i seguenti passi principali:

1. Inizializza i registri EAX e EDX con i valori 32 e 56 rispettivamente.
2. Somma questi due valori e memorizza il risultato in EAX, quindi EAX diventa 88.
3. Sposta il risultato in EBP e confronta EBP con 10.
4. Se EBP è maggiore o uguale a 10 (cosa che è vera in questo caso), salta all'indirizzo 0x1176.
5. Se il salto non avviene, imposta EAX a 0.
6. Chiama la funzione printf (per stampare il risultato)