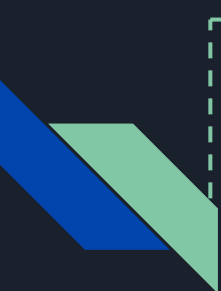


- 
1. ``mov EAX,0x20``: Questa istruzione carica il valore 32 (in decimale) nel registro EAX, che viene comunemente utilizzato per le operazioni aritmetiche.
 2. ``mov EDX,0x38``: Qui, viene caricato il valore 56 (in decimale) nel registro EDX, anch'esso utilizzato per memorizzare temporaneamente dati.
 3. ``add EAX,EDX``: Questa operazione somma i contenuti dei registri EAX e EDX e memorizza il risultato nuovamente in EAX.
 4. ``mov EBP, EAX``: Il valore risultante in EAX viene copiato nel registro EBP, utilizzato spesso come punto di riferimento per l'accesso a variabili locali e parametri delle funzioni.
 5. ``cmp EBP, 0xa``: Qui si confronta il valore nel registro EBP con il numero 10, preparando per un salto condizionale successivo.
 6. ``jge 0x1176 <main+61>``: Se il valore nel registro EBP è maggiore o uguale a 10, il programma passerà a un'altra parte del codice, indicata dall'indirizzo 0x1176.
 7. ``mov eax, 0x0``: Altrimenti, se il valore in EBP non supera 10, viene impostato il registro EAX a zero.
 8. ``call 0x1030 <printf@plt>``: Chiama la funzione printf, che di solito viene utilizzata per stampare output formattato su console. La stringa di formato e gli argomenti appropriati saranno presenti nello stack o nei registri prima della chiamata a printf.