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:

- ***mov EAX, 0x20:***

Questa istruzione muove il valore esadecimale 0x20 (32 in decimale) nel registro EAX.

- ***mov EDX, 0x38:***

Qui il valore esadecimale 0x38 (56 in decimale) viene caricato nel registro EDX.

- ***add EAX, EDX:***

Questa istruzione somma il contenuto di EAX (che ora è 32) con il contenuto di EDX (che è 56) e memorizza il risultato in EAX.

- ***mov EBP, EAX:***

Il valore risultante della somma precedente viene copiato nel registro EBP.

- ***cmp EBP, 0xa:***

Qui viene effettuato un confronto tra il contenuto di EBP (che è la somma precedente) e il valore esadecimale 0xa (che è 10 in decimale).

- ***jge 0x1176 <main+61>:***

Questa è un'istruzione di salto condizionale. Se il confronto precedente ha prodotto un risultato maggiore o uguale (jge = "jump if greater than or equal"), allora salta all'indirizzo 0x1176, altrimenti continua l'esecuzione dal punto successivo.

- ***mov eax, 0x0:***

Se il salto condizionale non viene eseguito, questa istruzione imposta il valore 0 nel registro EAX.

- ***call 0x1030 <printf@plt>:***

Infine, questa istruzione chiama la funzione printf, che presumeremmo stampi qualcosa a schermo, poiché è comunemente usata per la formattazione di output.