

LINGUAGGIO ASSEMBLY

0x00001141 <+8>: mov EAX, 0x20 – indica che il valore 0x20 (32 in decimale) viene salvato nel registro EAX.

0x00001148 <+15>: mov EDX, 0x38 – indica che il valore 0x38 (56 in decimale) viene salvato nel registro EDX.

0x00001155 <+28>: add EAX, EDX – somma il valore di EDX a quello di EAX e salva il risultato nel registro EAX ($32 + 56 = 88$).

0x00001157 <+30>: mov EBP, EAX – copia il valore contenuto in EAX nel registro EBP. EBP è la base della memoria stack, viene sovrascritta per iniziare una nuova operazione.

0x0000115a <+33>: cmp EBP, 0xa – compara il valore di EBP con il valore esadecimale 0xa (in decimale 10); dal momento che la differenza $88 - 10 = 78$, quindi maggiore di zero, lo zero flag e il carry flag rimangono inalterati.

0x0000115e <+37>: jge 0x1176 <main+61> – salto condizionale: se il valore della destinazione della precedente operazione cmp è uguale o maggiore della sua sorgente, il programma salta alla riga 61 della funzione main, contenuta all'interno della cella di memoria 0x1176 (440 in decimale). Equivale ad un ciclo if else.

0x0000116a <+49>: mov EAX, 0x0 – se la condizione del salto sopra è rispettata, viene assegnato il valore 0 al registro EAX, che viene quindi resettato.

0x0000116f <+54>: call 0x1030 <printf@plt> – se la condizione non è rispettata, il programma chiama un'altra funzione di stampa, alla cella di memoria 0x1030 (4144 in decimale).