

LINGUAGGIO ASSEMBLY

```
0x00001141 <+8>:  mov EAX, 0x20
0x00001148 <+15>:  mov EDX, 0x38
0x00001155 <+28>:  add EAX, EDX - somma tra le due variabili 32 + 56 = 88, le salva in EAX
0x00001157 <+30>:  mov EBP, EAX - copia 88 in EBP (base dello stack)
0x0000115a <+33>:  cmp EBP, 0xa
0x0000115e <+37>:  jge 0x1176 <main+61> - dato che EBP > 0xa saltiamo a cella di memoria 0x1176
                    (4470), dove si trova la riga 61 della funzione main, perché le condizioni del salto (maggiore
                    uguale) sono rispettate
0x0000116a <+49>:  mov EAX, 0x0 - sposta il valore 0 nel registro EAX (equivale a return 0)
0x0000116f <+54>:  call 0x1030 <printf@plt> - chiama una nuova funzione di stampa alla cella di
                    memoria 1030 (4144)
```

La prima linea indica che il valore 0x20 (32 in decimale) viene salvato nel registro EAX.

La seconda linea indica che il valore 0x38 (56 in decimale) viene salvato nel registro EDX.

La terza linea introduce una funzione somma tra i valori contenuti in EAX ed EDX e salva il risultato nel registro EAX ($32 + 56 = 88$).

La quarta linea salva il valore di EAX nel registro EBP, che è la base dello stack.

A questo punto inizia un ciclo if else; alla quinta linea l'operazione "compare" (cmp). La funzione dice di comparare il valore di EBP (88) con il valore esadecimale 0xa (10).

Alla sesta linea troviamo un conditional jump; se il valore di EBP è maggiore o uguale a 10 (0xa), allora il programma deve "saltare" alla riga 61 della funzione main, che si trova nella cella di memoria 0x1176 (4470 in decimale).

Alla settima linea abbiamo la chiusura del programma, quando il valore 0 è salvato nel registro EAX (equivale a return 0 del linguaggio C).

Se invece il valore di EBP fosse stato minore di 10, il programma avrebbe richiamato un'altra funzione, printf (di stampa), alla cella 0x1030 (4144 in decimale).