

# ASSEMBLY x86

**0x00001141 <+8>: mov EAX,0x20 :**

Sposta il valore esadecimale 0x20(32 in decimale) nel registro EAX.

**0x00001148 <+15>: mov EDX,0x38 :**

Sposta il valore esadecimale 0x38(56 in decimale) nel registro EDX.

**0x00001155 <+28>: add EAX,EDX :**

Fa la somma matematica dei valori presenti nei registri EAX e EDX andrà poi ad inserire la somma in EAX.

**0x00001157 <+30>: mov EBP, EAX :**

Sposta il contenuto del registro EAX nel registro EBP.

**0x0000115a <+33>: cmp EBP,0xa :**

Compara il valore esadecimale 0xa(10 in decimale), con il valore presente nel registro.

**0x0000115e <+37>: jge 0x1176 <main+61> :**

Il comando esegue un salto alla locazione di memoria 0x1176 se il confronto sopra darà come risultato che la destinazione è maggiore della sorgente .Nel nostro caso salterà.

**0x0000116a <+49>: mov EAX,0x0 :**

Sposta il valore esadecimale 0x0(0 in decimale) nel registro EAX.

**0000116f <+54>: call 0x1030 <[printf@plt](#)> :**

Richiama l' indirizzo di memoria 0x1030 in qui poi porta essere presente una funzione.