

# Assembly x86



## Codice completo:

```
0x00001141 <+8>:    mov   EAX,0x20
0x00001148 <+15>:   mov   EDX,0x38
0x00001155 <+28>:   add   EAX,EDX
0x00001157 <+30>:   mov   EBP, EAX
0x0000115a <+33>:   cmp   EBP,0xa
0x0000115e <+37>:   jge   0x1176 <main+61>
0x0000116a <+49>:   mov   eax,0x0
0x0000116f <+54>:   call  0x1030 <printf@plt>
```



**Commento:**

**0x00001141 <+8>: mov EAX,0x20** Assegna il valore esadecimale **0x20** (32 in decimale) al registro **EAX**.

**0x00001148 <+15>: mov EDX,0x38** Assegna il valore esadecimale **0x38** (56 in decimale) al registro **EDX**.

**0x00001155 <+28>: add EAX,EDX** Somma il valore nel registro **EDX** (56) con il valore nel registro **EAX** (32), e memorizza il risultato in **EAX**. Quindi, **EAX** ora contiene 88.

**0x00001157 <+30>: mov EBP,EAX** Sposta il valore di **EAX** (88) nel registro **EBP**. Ora, **EBP** contiene 88.

**0x0000115a <+33>: cmp EBP,0xa** Confronta il valore nel registro **EBP** (88) con **0xa** (10 in decimale).

**0x0000115e <+37>: jge 0x1176 <main+61>** Salta all'indirizzo **0x1176** (etichettato come **<main+61>**) se **EBP** è maggiore o uguale a **0xa** (10). Poiché **EBP** è 88, che è maggiore di 10, il salto avviene.

**0x0000116a <+49>: mov eax,0x0** Assegna il valore **0x0** (0 in decimale) al registro **EAX**.

**0x0000116f <+54>: call 0x1030 printf@plt** Chiama la funzione **printf** che si trova all'indirizzo **0x1030**.