

Traccia: Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. 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. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

```
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
0x0000116a <+49>:  mov  eax,0x0
```

0x00001141 <+8>: mov EAX,0x20 Sposta il valore dato dall' esadecimale 0x20 (**32** in decimali) nel registro **EAX** tramite l' istruzione **mov**.

0x00001148 <+15>: mov EDX,0x38 Sposta il valore dato dall' esadecimale 0x38 (**56** in decimale) nel registro **EDX** tramite l' istruzione **mov**.

0x00001155 <+28>: add EAX,EDX Fa la somma dei due valori decimali contenuti all' interno dei registri EAX ed EDX e lo sposta in EAX, in questo caso **32** e **56**. Quindi $32 + 56 = 88$

0x00001157 <+30>: mov EBP, EAX Sposta il valore contenuto nel registro EAX (88 dato dalla somma precedente) all' interno del registro **EBP**.

0x0000115a <+33>: cmp EBP,0xa Fa una comparazione tra il valore attuale all' interno del registro EBP (88, lo abbiamo precedentemente spostato dal registro EAX) ed il valore decimale dato dall' esadecimale 0xa (**10**)

0x0000116a <+49>: mov eax,0x0 Sposta il valore **0** (esadecimale 0x0) all' interno del registro EAX.