



Esercizio S10-L3



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
0x0000115e <+37>: jge  0x1176 <main+61>
0x0000116a <+49>: mov  eax,0x0
0x0000116f <+54>: call 0x1030 <printf@plt>
```



Svolgimento:

- 0x00001141 <+8>: `mov EAX,0x20`

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

- 0x00001148 <+15>: `mov EDX,0x38`

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

- 0x00001155 <+28>: `add EAX,EDX`

Somma i valori contenuti nei registri EAX ed EDX ed aggiorna il valore del registro

EAX col valore ottenuto dalla somma.

- 0x00001157 <+30>: `mov EBP,EAX`

Sposta il valore contenuto nel registro EAX all'interno del registro EBP.



Svolgimento:

- 0x0000115a <+33>: `cmp EBP,0xa`

Compara il valore contenuto nel registro EBP (88) con 0xa (10).

- 0x0000115e <+37>: `jge 0x1176 <main+61>`

Esegue un salto all'indirizzo 0x1176 quando il valore da comparare è maggiore o uguale al valore comparato, in questo caso la condizione è vera ed il salto si verifica.

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

Sposta il valore 0x0 (0) all'interno del registro EAX.

- 0x0000116f <+54>: `call 0x1030 <printf@plt>`

Richiama la funzione printf.