

## 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).

- 1) `mov EAX,0x20`
- 2) `mov EDX,0x38`
- 3) `add EAX,EDX`
- 4) `mov EBP, EAX`
- 5) `cmp EBP,0xa`
- 6) `jge 0x1176 <main+61>`
- 7) `mov eax,0x0`
- 8) `call 0x1030 <printf@plt>`

- 1) Muovi 0x20 nel registro EAX (0x20 in esadecimale è uguale a 32)
- 2) Muovi 0x38 nel registro EDX (0x38 in esadecimale equivale a 56)
- 3) Addiziona i valori nel registro EAX con i valori nel registro EDX, quindi 32 più 56 ovvero 88, quindi EAX equivale ad 88
- 4) Sposta/sostituisci il registro EAX nel registro EBP e quindi EBP acquisisce valore 88
- 5) Cmp è simile a Sub ma va a modificare soltanto i Flag, quindi avendo EBP con un valore di 88 ed un 0xa avente un valore 10 troviamo che la sorgente è minore della destinazione e quindi sia ZF sia CF avranno valore 0
- 6) Salta alla locazione specificata se la destinazione è maggiore o uguale della sorgente nell'istruzione «cmp» e quindi essendo questo il caso salta alla cella di memoria specificata tramite il puntatore 0x1176
- 7) resetta il valore di EAX a 0
- 8) chiama una funzione print del valore che sta nella cella di memoria il cui puntatore è lo 0x1030