

# ESERCITAZIONE W22D1



## Esercizio

Linguaggio Assembly

### 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>
```

3

Ecco le descrizioni per ogni riga di codice:

1. 0x00001141 <+8>: Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
2. 0x00001148 <+15>: mov EDX,0x38: Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
3. 0x00001155 <+28>: add EAX,EDX: Aggiunge il contenuto del registro EDX al registro EAX e memorizza il risultato in EAX.
4. 0x00001157 <+30>: mov EBP,EAX: Copia il contenuto del registro EAX nel registro EBP.
5. 0x0000115a <+33>: cmp EBP,0xa: Confronta il contenuto del registro EBP con il valore esadecimale 0xA (10 in decimale).
6. 0x0000115e <+37>: jge 0x1176 <main+61>: Salta all'indirizzo di memoria 0x1176 (etichettato come main+61) se il risultato del confronto precedente è maggiore o uguale.
7. 0x0000116a <+49>: mov eax,0x0: Copia il valore di EAX con il valore 0, ovvero copia 0 in EAX.
8. 0x0000116f <+54>: call 0x1030 <printf@plt>: Chiama la funzione printf situata all'indirizzo di memoria 0x1030.