

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>
```

- 1 - Spostamento variabile 32 (0x20) nel registro EAX
- 2 - Spostamento variabile 56 (0x38) nel registro EDX
- 3 - Somma il valore di EDX a EAX e salva il risultato in EAX
- 4 - Sposta il contenuto di EAX in EBP
- 5 - Confronta il valore di 10 (0xa) con il registro EBP
- 6 - Salta alla locazione 83 (0x1176) se EBP>=10
- 7 - Spostamento variabile 0 (0x0) nel registro EAX
- 8 – Istruzione printf@plt richiamata