

Oggi andremo ad analizzare diversi codici scritti in ASSEMBLY. Lo scopo è capire il loro funzionamento. I codici da analizzare sono i seguenti:

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>
```

1 codice: copia il valore 0x0,20 (32) nel registro EAX

2 codice: copia il valore 0x0,38 (56) nel registro EDX

3 codice: somma il registro EDX (56) al registro EAX (32)

4 codice: copia il contenuto del registro EAX (88) nel registro EBP

5 codice: compara il contenuto del registro EBP (88) con il numero esadecimale 0xa (10)

6 codice: usa il comando jge, ovvero effettuerà un salto alla locazione specificata ad una condizione: che la destinazione cmp sia maggiore del valore di controllo.

7 codice: copia il valore esadecimale 0 (0) nel registro eax

8 codice: chiama una funzione printf