

## S10 L3

Traccia: 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.

I numeri nel formato 0xYY sono numeri esadecimali e bisogna convertirli in numeri decimali.

```
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>
```

0x00001141 <+8>: mov EAX,0x20 = sposta il valore di 32 (0x20) nel registro EAX

0x00001148 <+15>: mov EDX,0x38 = sposta il valore di 56 (0x38) nel registro EDX

0x00001155 <+28>: add EAX,EDX = somma il valore di EDX a EAX,  $56+32 = 88$ , ed aggiorna il registro

0x00001157 <+30>: mov EBP, EAX = sposta il valore di EAX (88) nel registro EBP

0x0000115a <+33>: cmp EBP,0xa = compara il valore di 0xa (10) con il valore di EBP (88)

0x0000115e <+37>: jge 0x1176 <main+61> = effettua un salto di destinazione se cmp è >= . quindi 88>10, ed effettua il salto

0x0000116a <+49>: mov eax,0x0 = sposta il valore 0 (0x0) in EAX

0x0000116f <+54>: call 0x1030 <printf@plt> = chiama una funzione printf